Title: Identifying Benefits and Challenges in the application of Agile methodologies in software development

Name: Alistair Streek

Identifying Benefits and Challenges in the application of Agile methodologies in software development

by
Alistair Streek

Supervisor: Professor Yanqing Duan

A dissertation submitted to the University of Bedfordshire in accordance with the requirements of the degree of Master of Science.

March
2015

Abstract

Software has generally been developed using the ‘Waterfall’ methodology which is now believed to be cumbersome and slow to react to change. This belief has spawned new ‘Agile’ methodologies that aim to deliver small pieces of working software on a frequent basis. Many claims have been made about Agile, but little academic research has been carried out to justify them.

This study identifies the significance of benefits and challenges practitioners have found when adapting to Agile methodologies. Further, the benefits and challenges are linked to the Agile features in use and assesses what contributes to the level of user satisfaction.

A survey instrument was employed to collect data from as many practitioners as possible followed by post survey interviews. Claims made for Agile by service providers are compared to the experiences of those canvassed in the survey.

Results reveal that the most significant benefit is closer collaboration with other members of the development team. Significant challenges were the estimation of the time and effort Agile projects take to complete and that Agile only works with competent, motivated people.

The insights gained in this study will be circulated for the benefit of future research on software development methodologies and Agile practice.

Acknowledgements

I would like to express my appreciation to my supervisor Professor Yanqing Duan for her guidance and support.
My sincere thanks to everyone who took the time to complete one of my survey forms, or who submitted themselves to an interview.

To my wife for her patience and support and to my mother without whose assistance this would not have been possible.

Declaration

This dissertation is submitted to the University of Bedfordshire in accordance with the requirements of the degree of Master of Science. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Alistair Streek

February 2015

Contents

List of figures

List of Tables

Chapter 1. Introduction

1.1 Overview

Agile is always described by its advocates as being a far more efficient way to create software (Sliger & Broderick, 2008; Messenger, 2014). The claim is that Agile will improve productivity by doing away with detailed requirements and specification documents and replace them with a collaborative approach with very little documentation and where there are no all-encompassing decisions made at the beginning of the project. (Beck et.al., 2001)

There has been little investigation of the problems with using Agile (Papatheocharous & Andreou, 2014; Abrahamsson et al, 2002) and the drawbacks to using the methodology. In fact in most organisations, Agile is promoted with an evangelical zeal and questioning how the methodology could be adapted or improved is not encouraged (Shahir, 2008). This is demonstrated by any challenge to process being held as an example of the plaintive 'not using Agile properly'. When methodologies are being discussed and Agile is the preferred method of delivery there is a view that there are only two ways to implement software, Waterfall or Agile (Mnkandla, 2008; Conboy, 2009). In practice it should be possible to combine elements of both.

Using the 'big specification' Waterfall methodology is risky because after investing a significant amount of effort into a detailed and comprehensive design, the project may fail to meet users' needs (Cohn, 2004). There are many reasons why this might be the case, for example the wrong technology may have been specified, or after a large section of the development has been completed it might be discovered that the design will not meet the needs of users. During the life of a large project which might take months or years, the requirement originally met by the design may have changed (Mellor, 2005).

Agile is designed to use short development cycles or 'Sprints' of between one and four weeks (Cohn, 2004). After each sprint a deliverable can be demonstrated to users or analysts, which means that the risk that the design is wrong and will have to be discarded is mitigated (Leffingwell, 2010).

By using the Agile approach there are some inherent weaknesses; for example, there is no requirement for a master document which explains how the deliverable will work end-to-end (Cho, 2009). Likewise, the concurrent nature of Agile does not allow for delays between analysis, development and test.

From personal experience of Agile it is apparent that there are issues for development teams, but by taking a broader perspective from a significant number of users of Agile, it is possible to provide a better basis for investigating what difficulties are encountered and how those difficulties can be met.

There is little reliable evidence of where Agile users feel that the methodology really works well. This study will explore where users feel Agile is weakest and strongest by using questionnaires and personal interviews. Data analysis will highlight areas of the methodology which are seen as difficulties, enabling Agile teams to adjust their processes to mitigate the issues (Cockburn, 2003, Fitzgerald & Hartnett & Conboy, 2006).

1.2 Aims and objectives

Previous studies have argued that the majority of the very few empirical studies of Agile are descriptive and lacking data on how users perceive Agile methodologies (e.g. Wang et al, 2012; Lee and Xia,2010; Dybå and Dingsøyr, 2008; Erickson et al, 2005; Conboy, 2009; Barlow, 2011; Hong et al, 2011).

The aim of this study is to identify benefits and challenges in the application of Agile software development methodology from a user's perspective.

More specifically, the research has the following objectives:

1. Understand the current issues surrounding the agile application through literature review.
2. Understand how Agile users are using Agile features, their level of satisfaction and perceived benefits and challenges by collecting empirical evidence from Agile practitioners in different organisations.
3. Identify areas of specific Agile methods such as Scrum and Kanban that are seen to be problematic. Then to understand how users are addressing commonly found issues.
4. Raise awareness and disseminate the findings for the benefit of the wider Agile community.

To achieve the research aims objectives this study will undertake the following activities:

1. Conduct a literature review to understand the current issues surrounding the agile application.
2. Canvas Agile professionals for empirical investigation.

3. Develop research questions and design instruments for data collection. The data will then be used to assess the level of agile adoption in the users' organisation and their opinions on the benefits and challenges of using an Agile methodology.
4. Analyse the data and summarise the key findings.
5. Identify areas of Agile that are seen to be problematic and select appropriate users for interviews to find out how they are addressing commonly found issues.
6. Analyse the data and summarise the key findings.
7. Disseminate the findings for the benefit of the wider Agile community.

1.3 Why investigate Agile?

As a Business Analyst working in an Agile team, I started to use Agile three years ago and read many articles on the internet about the use and practice of Agile. Some of the providers of Agile services and consultancy made claims about the practice which seem over optimistic and unfounded. For example "Agile teams are more productive than teams using traditional methods".

This prompted me to dig deeper into Agile – initially I looked for empirical data on Agile productivity, but there did not appear to be any as it is extremely difficult to measure whether a team is producing more useable features and what the quality of the output is without making assumptions which invalidate the analysis.

If it is not practically possible to test the amount of code produced under Agile, I was expecting to find investigations of what users find useful and difficult about development. Again there appeared to be very little empirical data available. As a practitioner, I felt that information about user perceptions would be extremely useful as an aid to our development, which prompted this research.

1.4 Research method

The research was conducted using a pragmatic approach, a mixed research method both qualitative and quantitative were employed to meet the objectives of the study.

Because Agile is a new and rapidly changing environment, an interpretive epistemology has been employed (Walsham,1995).

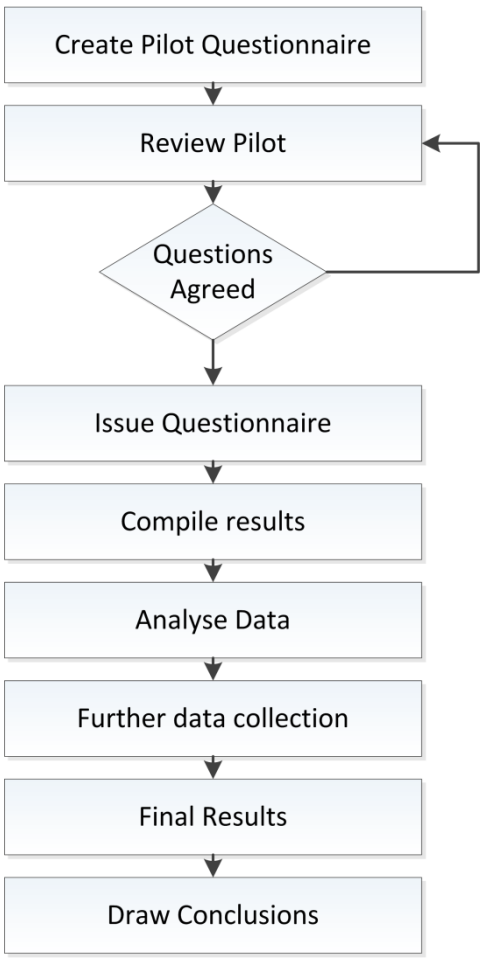

Figure 1-1 Research Method

From personal experience initially, followed by discussion with colleagues and three experienced Agile practitioners from other organisations, a pilot questionnaire was produced. This was then piloted with several groups of Agile practitioners until a final questionnaire was developed.

The questionnaire was then handed out in hard copy and reproduced as an on-line survey using the Qualtrics survey tool. A variety of methods of obtaining responses were used.

The responses were all collected on Qualtrics, then the data was analysed using SPSS.

Where the original finding produced unexpected or significant results which required further investigation, additional questions were asked to a sub-set of the original questionnaire respondents.

Finally the data was summarised and conclusions drawn.

1.5 Dissertation outline

The dissertation consists of five parts:
Chapter 2 refers to the background and concepts of Agile.
Chapter 3 the literary review
Chapter 4 presents the methodology employed
Chapter 5 describes data analysis
Chapter 6 conclusion

A critical evaluation of the project and final conclusions are provided in Chapter 6. Trying to be as objective as possible and discussing to what extent objectives have been achieved.

Recommendations for further work are drawn up in chapter 6.

Chapter 2. Background and concepts of Agile

2.1 Introduction

This chapter aims to describe the methodologies used for software development and their evolution. By far the most common way that software has been developed in the past has been the ‘Waterfall’ method (Holtsnider et.al., 2010). This will be explained to give a common understanding of process and terminology.

Software development is prone to changes of fashion in the same way as any other industry. A new methodology ‘Agile’ has become popular and its usage continues to increase (VersionOne, 2014). This methodology will be described to give a common understanding of what users of Agile are trying to achieve.

The aim of this section is to place the research in context within the changing world of software development.

2.2 Waterfall Software Development

The Waterfall method of development has been used since the 1960s with varying degrees of success. It relies on a sequence of activities, one after the other to manage and document the software design process.

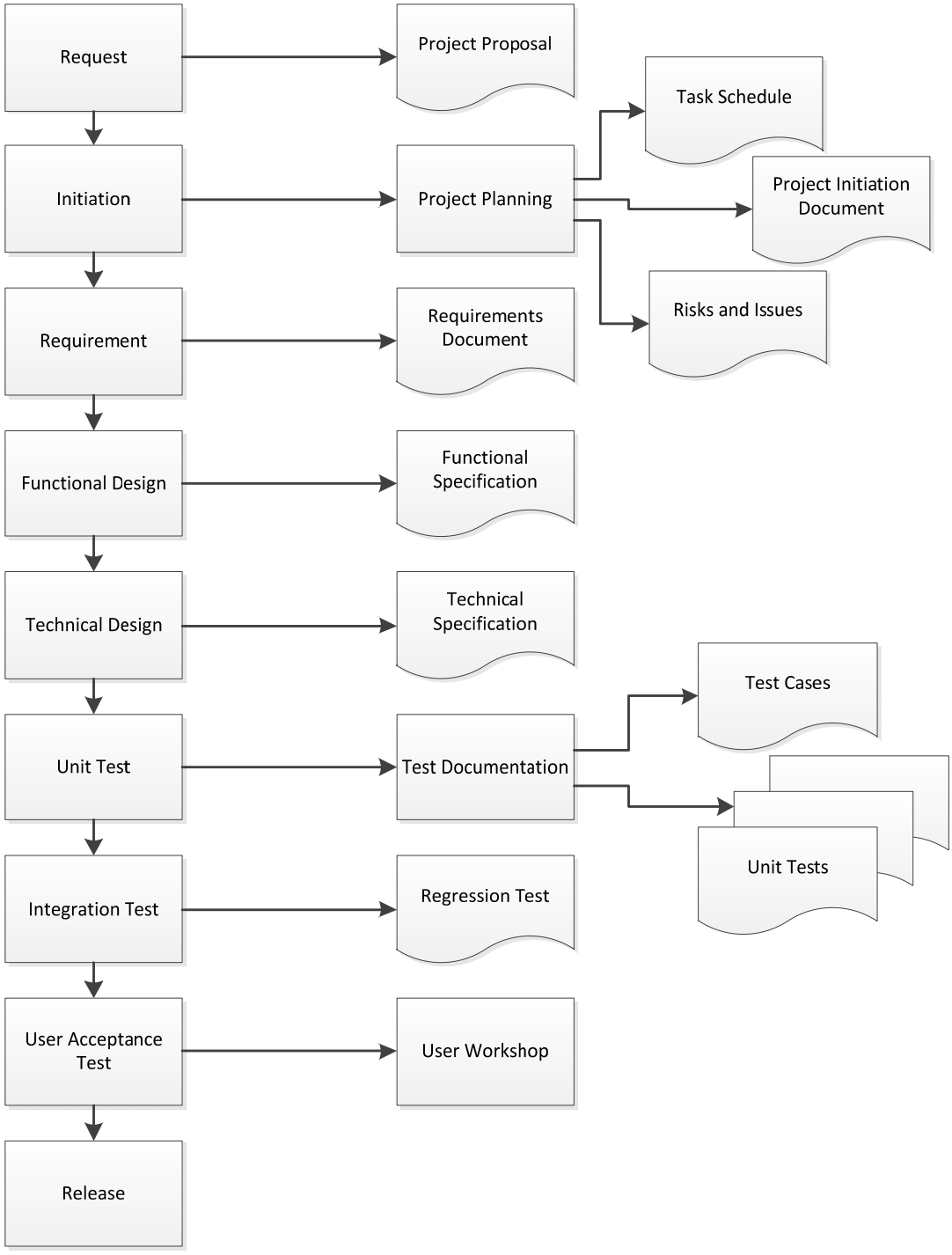

Figure 2-1 Waterfall Process (Guntamukkala et. al., 2006; Chauham, 2003)

After each of the first four steps in this process, the documentation which has been created is agreed and signed off, thereby committing everyone to the next stage of the process (Royce, 1970).

Each step is performed by a specific role within the project.

Development Request – Project Manager

Take a request from the owner of the product, or directly from customers. Identify the purpose and scope of the development.

Project Initiation – Project Manager

Plan the project, identify team members and commit resources to the project. To mitigate risks and agree success criteria.

Requirements Analysis – Business Analyst

Create a detailed requirements document from the initial request for development.

Functional Design – Business Analyst

Write a detailed description of how the new process or processes should work, define inputs/outputs and validations.

Technical Design – Technical Analyst or Developer

Expand the Functional Design to define tables, technical definitions for fields and validations.

Development – Developer and Documentation

Build the programs and processes.

Unit Testing – Tester

Test the new programs and processes in isolation, using the Functional Design as the base document.

Integration Testing - Tester

Test the new programs and processes in a complete environment, ensure that no other part of the software is compromised by the new code.

User Acceptance Test – Users

The new programs and processes are tested by end-users, using the requirements document as the base document.

Release

Package the new programs and processes into a releasable format and issue it to end users.

2.3 Agile Software Development

The Waterfall methodology has been used to develop software for many years and while this has succeeded in producing working software, it has been seen as prescriptive and unresponsive by proponents of Agile (Barlow et. al., 2011) and other Rapid Application Development methods (Martin, 1990).

The features of the Waterfall method which Agile seeks to address are:

- Waterfall does not promote different disciplines working together at all stages of the project.
- It is possible that a Waterfall project could get as far as the code development before the project team realise that what is being asked for is not technically possible.
- A large amount of documentation is produced, much of which will never be viewed after the project is complete.

2.4 The Transition to Agile

Waterfall development has been very productive for the last forty years, but there have been several attempts to introduce other methodologies. A brief overview of the most popular of these methodologies will be covered in this section.

2.4.1 Spiral

A derivative of Waterfall is the spiral development methodology wherein elements of design and prototyping are combined, using both top-down and bottom-up design (Boehm, 1995).

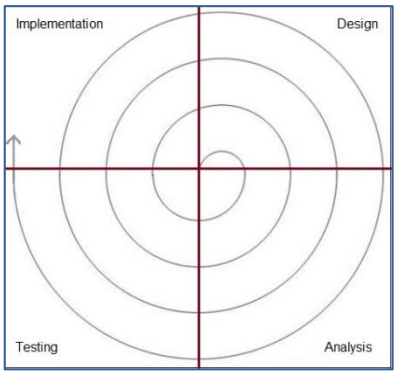

Figure 2-2 Spiral Development (Boehm, 1995)

In the spiral model, risk is reduced by splitting the project into sections. Each section is

then developed, creating a prototype while building upon previous iterations. Then the next section of code is started, with each iteration comprising the same four steps; design, analysis, testing and implementation.

2.4.2 Object Orientated Development

Object Orientated Development (OOD) was used from the 1980s onwards and is sometimes described as a methodology. However it is essentially a waterfall development process which bases the delivery of code not on a single large deliverable of code and one block of data which the code acts upon, but as ‘objects’, each of which had code and data attached to them. OOD is very much how the code is delivered rather than a complete methodology (Booch, 1986).

2.4.3 Rational Unified Process

Another method much discussed in the late-1990s is Rational unified Process (RUP) which is an iterative method based on a component-based architecture (Khan et. al, 2011). RUP was originally developed as a software process product by a company called Rational, prioritises its iterations based on the priority and the risk of each feature. While the methodology is iterative it does require a formal project plan defining what is to be created within each iteration.

2.4.4 Rapid Application Development

The Rapid Application Development (RAD) methodology was much talked about at the end of the 1990s and was based on the use of small iterative developments using Computer-Aided Systems Engineering (CASE) tools. Certainly this can be seen as a forerunner of Agile, though it was based on prototyping rather than creating small sections of requirements which are delivered iteratively.

The most prominent proponent of RAD was James Martin (Martin, 1990) who formalised the RAD methodology. As described by Martin “RAD refers to a development lifecycle designed to give much faster development and higher-quality results than those achieved with a traditional lifecycle. It is designed to take maximum advantage of powerful development software that has evolved recently.”

While RAD was much discussed by practitioners, it never produced enough momentum to become a generally used process.

2.5 Agile Development

Designed to be able to react more quickly to changing circumstances, Agile was formalised in 2003 by a group of Agile service providers who defined a group of development principals in the Agile Manifesto. (Beck et. al., 2001)

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over Processes and tools

Working software over Comprehensive documentation

Customer collaboration over Contract negotiation

Responding to change over Following a plan

That is, while there is value in the items on the right, we value the items on the left more."

From these values twelve Agile principals were agreed and from these a number of development methodologies have been created which provide a framework for software development. All of the Agile methodologies promote teamwork, collaboration, iterative development and the removal of unnecessary documentation and process from projects (Mnkandla, 2008).

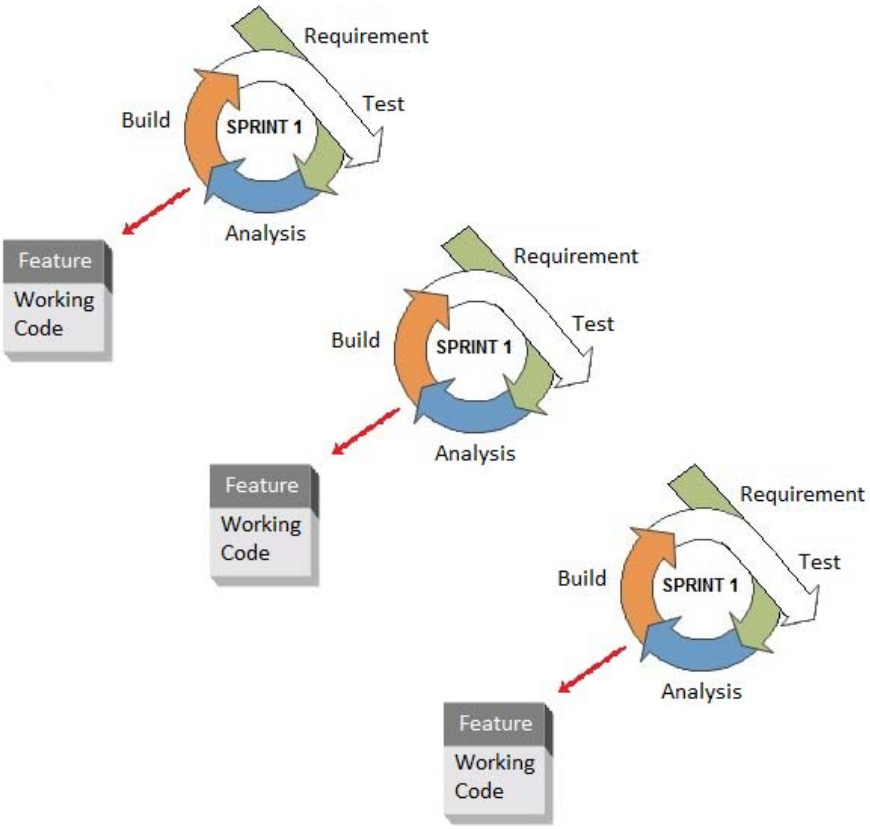

Figure 2-3 Agile Development (Reich, 2012)

Agile consists of a series of small cycles, each of which consist of analysis, development and testing.

Agile breaks requirements into small pieces, each of which is estimated just in time for the next development sprint, so that time is not spent unnecessarily planning work rather than executing it. Agile teams are conditioned not to think in terms of hours of work, but in terms of deliverables (Cohn, 2004).

2.6 Overview of Agile Methods

To place the research in context Scrum and Kanban will be elaborated upon, as these are the focus of this research.

2.6.1 Scrum

The word scrum is taken from Rugby Union and refers to re-starting a game after infringement. Within the Scrum methodology, work is broken down into small sections called Stories. Stories are held in a prioritised list called a Product Backlog. The Backlog contains features, bug fixes and any other work which is required to deliver the software (Schwaber & Beedle, 2002).

Scrum prescribes four formal events for inspection and adaptation:

- Sprint Planning
- Daily Scrum Meeting (a fifteen minute meeting to synchronise activities and plan for the next twenty-four hours).
- Sprint Review
- Sprint Retrospective

Scrum Teams are self-organising and cross-functional. The idea behind this being that self-organising teams choose how best to accomplish their work. Cross-functional teams have all the competencies required to deliver working code without depending on others who are not part of the team.

Work is delivered in short bursts called Sprints. Sprints can be between one and four weeks and the anticipation is that at the end of a Sprint a working and potentially shippable product has been created (Cohn, 2004).

Scrum depends on a number of roles within the development team:

- Product Owner who defines requirements and works with the team to identify and prioritise the work.
- The Scrum-master manages the Scrum process.
- But most of all the team. Teams work together to decide on what would be a potentially shippable product. The team then signs up to deliver that work within the Sprint. Once the deliverables in a Sprint have been agreed, no additional work can be added to the Sprint without the agreement of the team.

2.6.2 Kanban

In Japanese Kanban means “signal card” and the methodology uses a board to schedule tasks on a just-in-time basis (Anderson 2010). In a manufacturing environment nobody in a Kanban team is allowed to start work on a task unless it has been passed on to them from the previous worker.

The Kanban Method is used by organisations to manage the creation of products with an emphasis on continual delivery while not over-burdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively. Kanban uses Work Items, which in practice are usually User Stories, again similar to Scrum.
Work Items again are placed in a prioritised backlog (Stellman and Greene, 2014).

Kanban is based on 4 principles (Hurtado, 2013):

- Visualise what you do today
- Limit the amount of work in progress (WIP) which helps teams to avoid starting and then committing to too much work at once
- Workflow where when something is finished, the next highest thing from the backlog is pulled into play
- Kanban promotes continuous collaboration and encourages ongoing learning and improving process by defining the best possible team workflow.

In Kanban there are always work items in progress. Releases are planned and whatever work has been completed when a release is reached in included in that delivery.

2.6.3 Scrumban

A recent innovation Scrumban combines the Scrum and Kanban methodologies (Ladas, 2009);

- Use the prescriptive nature of Scrum to be Agile.
- Use the process improvement of Kanban to allow the team to continually improve its process.

By using the basic Kanban process, software is developed and at any point, there is always work in progress. As the team gets closer to the release of software, a decision point is reached where no more work is added to the work in progress. Ultimately the release contains all of the work items, with none remaining in progress.

2.6.4 Other Agile methodologies

Though not the focus of this study, there are other Agile methodologies worthy of note since there are elements which are sometimes incorporated into Scrum and Kanban teams:

Lean - primarily designed to eliminate waste through such practises as selecting only the truly valuable features of a system, prioritising the selected features and delivering in small sections (Poppendieck and Poppendieck, 2003).

Crystal – created by one of the signatories of the Agile Manifesto, Alistair Cockburn. Crystal is based on flexibility and the premise that different projects call for different methodologies (Cockburn, 2004).

DSDM - developed in 1994 and is the oldest of the popular Agile networks. DSDM is primarily a project management tool and is generally used in conjunction with other iterative methodologies such as Scrum and XP (Stapleton, 2003).
DSDM promotes the use of Moscow rules (MSCW) for prioritising work:

M – Must have requirement

S – Should be delivered if at all possible

C – Could be delivered but not critical

W - Won't be delivered in this iteration

XP (Extreme programming) – focused on development, advocating frequent "releases" in short development cycles. XP is based on four simple values; simplicity, communication, feedback and courage. 'Courage' is mentioned in other methodologies, but is a key feature of XP (Lindstrom and Jefferies, 2004). Rapid lean development used by XP means that documentation is very limited.

FDD (Feature Driven Development) - consists of a series of two-week "design by feature, build by feature" iterations. Features are small customer deliverables. FDD practices are suited to using development tools and the methodology is designed to scale up to very large projects easily (Palmer and Felsing, 2001).

2.7 Features of Agile Development discussed during analysis

2.7.1 Stories and pointing

A story represents a small feature or block of work. Stories should be testable and deliverable to users.

There is no single way to write or manage stories, but Mike Cohn in his book “User Stories Applied” (2004) suggests that stories should be Independent, Negotiatble, Valuable to users, Estimable, Small and Testable.

Rather than work hours stories are given points. Points show the relative effort of one story to another and points are estimated for each feature, based on the Fibonacci sequence.

A total number of points for the whole project can be obtained, based on the previous speed of development the work involved for that phase of development can be calculated

2.7.2 Task Radiator

The ‘task radiator’ or ‘card wall’ is a tool for tracking the progress of stories (Al-Baik and Miller, 2014).

Figure 2-4 A task radiator or card wall (Hiranabe, 2007)

Again there is no ‘standard’ way to set up the radiator, but a common feature of all of them is the movement of stories from ‘To Do’ on the left to ‘Done’ on the right (Anderson 2010).

The columns in between can split work between development and test. Sometimes boards have a column for ‘Ready’ advising that stories have been developed and tested but are not ready to be flagged as ‘Done’. Stories in ‘Work in Progress’ will generally have a mark attached to them showing who is currently working on them.

Any story which cannot be progressed can be marked as blocked and can be escalated for the development or project manager responsible for delivery’s attention.

The Task radiator allows anyone walking past the development team to see their progress and who has been allocated to each task.

2.8 Author's Standpoint

The view of how these methodologies work is based upon that of the observer and in order to judge my perspective I should explain my background with development methodologies.

I have worked in software development for twenty-five years, most of that time as a Business Analyst though I have also been a developer, tester and Project Manager. Working originally on Waterfall projects with a brief venture into RAD in the late 1990s, in July 2011 I started working with Agile.
As a practitioner, I am focused on use and enhancement of Agile rather than forming a theoretical model. This is reflected in the emphasis placed on my research.

2.9 Summary

This chapter provides the general background of the project and analyses the basic theory that will support the methods. Agile methodologies based on small incremental developments are replacing the more prescriptive Waterfall development.

Chapter 3. Literary Review

3.1 Introduction

This chapter provides an overview of current studies and debates in relation to the benefits and challenges of Agile methodologies. The insights gained from the existing research will help to identify the gaps that will inform the research design of the present study.

The process of this review includes:

- Identifying any prior work which relates to the user's perceptions of Agile, concentrating on those in high-impact journals, but also encompassing any other relevant studies, conferences or surveys.
- Reviewing Agile case studies which touch on user satisfaction.
- Looking at the current Agile literature in a wider sense by investigating books, articles and any other source materials which comment on Agile without concentrating on user perceptions.

To assess the information already available about Agile, a comprehensive review of existing literature places this study in context. Data has been obtained from as many sources as possible to verify that that this research has not already been performed and to provide justification for its conclusion based upon previous research. More emphasis has been placed on studies published in high-quality academic journals.

There have been few studies of Agile in high-quality journals bearing in mind the large number of projects which use the methodology and the increase in its use year on year. In particular the variants of Scrum and Kanban have not been covered because Academic studies (Dybå and Dingsøyr, 2008) concentrate on Extreme Programming XP, which is particularly popular within Universities' IT departments.

Agile methodologies are increasingly popular and according to the Version One (Versionone, 2014) sponsored '2013 State of Agile Development survey' 57% of respondents said their companies have adopted Agile practices across 5 or more teams. This number has increased from 48% in 2012 and 33% in 2011. The survey canvassed 3501 individuals working within the software development industry. According to the survey 66% of organisations using Agile, employ the Scrum or Scum XP hybrid method with a further 12% using Kanban or Scrumban. For this reason data within this study has been collected for Agile users who employ Scrum and Kanban, as the methods are

broadly similar and cover 78% of the Agile community.

Since the research within this thesis concentrates on the use and improvement of Agile rather than the conceptual framework, the literary review will likewise concentrate on the usage and improvement of Agile.

Agile literature has been evaluated by Dybå and Dingsøyr (2008) though their article is now six years old. A further review was carried out by Lee and Xia (2010) and PhD theses investigating Agile also contain exhaustive literature reviews (Aswani, 2008; Abbass, 2009).

It is difficult to set a quantitative measure on how effective any process or methodology is, when looking at software development (Brooks, 1980) since it is difficult to obtain a baseline. Even with a baseline, the work of a development team has to be assessed where there are no external factors which might affect the development process, such as change in staff members, or roles within the team. Quite apart from the difficulties of accessing a company's data, obtaining a reliable measure is subject to many other factors.

Case studies of how Scrum is used (e.g. Vlaanderen et al, 2011; Moe et al, 2011) of team interactions (e.g. Moe et al, 2009) and practitioner reviews of Agile strengths and weaknesses, for example by Shahir, Daneshpajouh and Ramsin (2008) have been performed. However, after a rigorous search of high-impact academic journals, no empirical studies of the Scrum or Kanban methods have been found for a large sample of users. One study does compare an XP method Agile team with a similar sized waterfall team and highlights the benefits and difficulties of using Agile (Vidgen and Wang, 2009) but this is not directly applicable to Scrum/Kanban and is derived from only two case studies.

Empirical research (e.g. Wang et al, 2012; Salo and Abrahamsson, 2008) and theoretical research (e.g. Chan, 2008; McAvoy and Butler, 2009) has been performed on the adoption of Agile processes and the studies have all suggested that additional research is required to give Agile a stronger academic justification.

3.2 Development of Agile

The first record of decentralised programming was made in the nineteen-fifties (Bennington, 1956) wherein the breaking down of a large program into sections each to be written by small teams was suggested.

Large-scale use of incremental programming was initiated by Rapid Application Development (RAD) which was a popular subject for software practitioners in the early nineteen-nineties (Rettig 1993). However, while RAD never took off in the way that Agile has (Howard, 2002) there are many similarities between RAD and Agile (Goth, 2009). Like Agile, RAD concentrates on small projects, collaboration and developer empowerment, though to use the methodology these must be used in conjunction with CASE tools to automatically generate complex code from simple instructions (Martin, 1991). When RAD was proposed it was often seen as being an excuse for ignoring software standards and producing low-quality software (Howard, 2002) and did not become widely adopted. Very few academic studies of RAD have been made (Beynon-Davies et. al., 1999; Beynon-Davies, Mackay and Tudhope 2000) so it is difficult to quantify its effectiveness accurately.

The style of development, now referred to as Agile was popularised by practitioners in the late nineteen-nineties and formalised in 2001 as the Agile Manifesto (Beck et. al., 2001)

While there are very few empirical studies of Agile in high impact journals, (Laanti, Salo and Abrahamsson, 2009) there are a great many non-academic articles describing Agile usage on websites and a large number of books written by practitioners.

When assessing the effectiveness of development practices, articles generally compare Agile to purely Waterfall methods and as pointed out by Batra et. al. (Batra et. al., 2010) they do not consider a hybrid use of both methodologies. Within Agile it is also difficult to identify exactly which method is being used, historically as suggested by Fitzgerald (Fitzgerald, 1997) development teams have not fully embraced methodologies. In addition to this, many teams use elements of different Agile methodologies tailored to their own needs (Pikkarainen, Salo and Kuusela, 2012).

These factors make it more difficult to assess how Agile is perceived, as there is no single methodology to consider.

3.3 Agile Benefits

Benefits are claimed for Agile development by service providers, usually those who provide consultancy services to the software industry often without any evidence. The benefits claimed by those with a vested interest in convincing potential customers to employ Agile methods are sometimes also described by more credible research, both academic and within private research organisations.

3.3.1 Claims for Agile by Service Providers

Agile manifesto principals are claimed to be delivered by consultants and service providers using different combinations of processes from Agile methodologies. They also make specific statements about the benefits of Agile, while avoiding the issues that teams face when using an Agile methodology.

Similar claims are made by a number of agile consultants -

"all about Agile" (Waters, 2012), Version One (2013), Mark Layton (Layton, 2012) and Seguetech (Zolyak, 2013) IBM (Ekas, 2012):

- Deliver working software early.
- Early and regular releases.
- Early visibility of any quality issues.
- Early visibility of project issues.
- Greatly enhanced ability to embrace change.
- The cost of completing each iteration is more predictable.
- Much better stakeholder engagement.
- Significantly reduced project risk.
- Customers are more satisfied by the deliverables.
- Higher team morale.
- Better visibility of project progress.
- Better product quality.
- Always have a working product.

These claims are backed up by some of the most popular books about Agile (Cohn, 2004; Leffingwell, 2011; Anderson 2010) and while the books have been used as guide for Agile teams they are also written by Agile consultants rather than academics.

Some providers make more speculative statements for example Seguetech claims:

“Solutions delivered in up to 50% less time.”
IBM claims:
“Minimal functionality does not imply poor functionality.”
“Agile teams are more productive than teams using traditional methods.”
“Using Agile streamlines and reduced project overheads.”
Within the book ‘Agile Project Management for Dummies’ (Layton, 2012):
“Agile Embraces technological excellence.”

These claims seem to be based solely on the consultants’ opinion and have no foundation in empirical data. Agile is often sold by service providers as a panacea and the benefits of its use are over-claimed. While stakeholders have sometimes been canvassed on a convenience basis, users’ views have not been included in the articles they have published.

3.3.2 Agile Analysis by Microsoft Research

The most relevant analysis of issues and benefits has been performed by Microsoft Research (Begel and Nagappan, 2007) and while the data collection was limited to one company (Microsoft) and was not conducted with the academic rigour required by a major journal, the results are still significant. A total of 488 responses were received. Of these only 156 of the respondents were using Agile, of which 65% use Scrum.

Around 60% of the users of Agile said that Agile is working well for them, but few of the respondents believe that Agile is working less well than their previous methodology.

The top six benefits of Agile were:

1. Improved communication and co-ordination
2. Quick release of working software
3. Short sprints allow flexibility of design
4. Developers waste less time on tasks they perceive as irrelevant
5. Improvement in software quality, manifested by having fewer bugs
6. Better focus on Customers

Respondents were asked to say what they liked and disliked about Agile as an open question. Again they re-enforced the finding of the survey by rating improved co-ordination between team members, quick release of deliverable software but disliked the use of Agile for projects with more than twenty team members and the excessive overhead of having to attend too many meetings.

No attempt has been made within this study to link benefits or challenges to the Agile features used by the respondents to the survey.

3.3.2 Review of Academic Articles

3.3.2.1 Examination of Agile survey-based studies

An article has been produced by de Cesare et. al. in the Communications of the ACM (de Cesare et al, 2010) examining the perceptions of Agile. While the study is more focused on the perception of Agile principals rather than the analysis of benefits found by users of Agile, it is relevant. The research was directed at IT directors, Senior Architects and Managers, rather than to team members.

The principals of Agile that more that 50% of users deemed to have high importance are:

1. Business people and developers work together throughout the project.
2. Frequent delivery of working software.
3. Face to face conversation is the most effective method of conveying information to the team.
4. Projects should be built around motivated individuals who have the support they need to succeed.

Delivery of working software is seen as important, but from a project perspective working software is only seen as the primary measure of progress in a quarter of projects. Project managers still use the same project metrics as they did on Waterfall projects and base their progress on milestones, large deliverables or completion of lifecycle stages.

It was found that 82% of organisations which use a formal development process heavily customise them.

While Agile adherents promote reduced documentation, the article finds that 50% of its respondents view the documentation generated by their project as necessary and sufficient, with a further 38% believing that more documentation is required.

The primary weakness in this analysis is that the sample group are not team members. Again there is no attempt to correlate the results with the features used by the Agile teams.

A further analysis of user's perceptions of Agile has been performed by Ani Asnawi in her PhD thesis from Southampton University (Asnawi, 2012).

The thesis concentrates on the adoption of Agile in Malaysia and while it concentrates on awareness and adoption of Agile, it also canvasses the opinions of practitioners as to the perceived benefits and challenges of Agile. The PhD findings are summarised in the conference paper subsequently submitted to the Agile India conference in 2012 (Asnawi, Gravell and Wills, 2012).

The key findings of this research are:

- Companies interviewed have adopted Agile as a consequence of bad experiences in Waterfall development. It is apparent from the research that companies only partially converting to Agile do not receive the full benefits of incremental development.
- Companies fully committed to Agile methods have positive perceptions of Agile.
- The most significant benefit of Agile is the involvement of all parties in the development process.
- The daily stand-up meeting was seen to be a significant benefit.
- It is easy for team member to track the progress of their projects under Agile.

While this thesis does cover benefits and challenges users find while using Agile, it is not the primary area of research, which concentrates far more on adoption rather than user perspective.

A third study has been performed by Vijayasarathy and Turk (2008) which obtained data through an anonymous on-line survey, the study group being contacted through online user groups. 98 completed survey forms were received. One section of the survey asked about the challenges found while using agile and the benefits realised. Only seven challenges and six benefits questions were asked listed in the survey.
Of these the most significant challenges were:

- Organisational resistance
- Management apathy
- Inadequate training

And benefits:

- Better meets customer needs
- Improved software quality
- Increased flexibility in development

Unfortunately no sample questionnaire was included in the article.

3.3.2.2 Examination of Agile Case Studies

Many case studies about Agile have been published, though few of them deliver specific information on the benefits and challenges of using Agile.

Of the case studies of how Scrum and Kanban are used which have been published that consider the issues and benefits of Agile, the ten most relevant (Schatz and Abdelshafi, 2005; Fitzgerald, Hartnett and Conboy, 2006; Smith and King, 2008; Laanti, Salo and Abrahamsson, 2009; Moe et al, 2009; Barlow et al, 2011; Vlaanderen et al, 2011; Spence, 2005; Overhage and Schlauderer, 2012; Abbas, 2009) draw conclusions about the use of Agile:

- Daily meetings are essential.
- An Agile environment is dynamic and is constantly adapting.
- Agile Methods focus on customer needs.
- Agile is generally popular with its users particularly with small or medium sized teams.
- Agile promotes collaboration.
- Different variations of Agile should be used for different teams, as what suits one team is unlikely to be right for another.
- Improved time to market.
- Agile is more enjoyable the traditional methods.

Studies have been published which examine how Agile works for example that by Shahir, Daneshpajouh and Ramsin (Shahir et al, 2008) but these articles concentrate on the creation of a framework by which Agile can be measured rather than analysis of data received from users.

Agile does appears to be popular with the majority of its users, but there are questions over its suitability for large complex projects. Batra et al. found in their case study that Agile can be used successfully for large projects in combination with a structured method. (Batra, 2010).

3.3.3 Common Benefits from Literature

Features brought up in the various studies tend to fall into two categories, people-related ‘soft’ aspects and methodology-related ‘hard’ aspects (Conboy and Coyle, 2011; Dybå and Dingsøyr, 2008, Misra, Kumar and Kumar, 2009; Chan and Thong, 2009). Benefits have therefore been broken down into those types.

Further to this Dybå and Dingsøyr break the Methodology feature further into ‘Development’ and ‘knowledge/project management’ groups (or themes).

This split into development and methodology processes also reflect the groupings of the principals of Agile specified in the Agile Manifesto suggested by Sidkey and Arthur (2007) ‘plan and deliver software frequently’ and ‘technical excellence’.

The Methodology section has therefore used a breakdown into two sections which will be described as:

Methodology	Process Management
	Speed of Release

People factors, like those for methodology can likewise be broken down a little further. Dybå and Dingsøyr again break the factor into themes ‘Collaboration’, ‘Organisational culture’ and ‘Team characteristics’. De Cesare et al broke the people factors into ‘Communication and Collaboration’ and ‘Team Involvement’ (de Cesare et al, 2010). For the purpose of this research only two themes ‘collaboration’ which appears throughout the Agile literature and ‘Team activities and integration’ will be used. Again the benefits described in the literature have been broken into these also shown in the Themes column.

Chan and Thong (2009) break down the people factors into ‘Ability related’, ‘Motivation related’ and ‘Opportunity related’ factors. The Opportunity related factors are all related to collaboration and Ability and Motivation both relate to personal and team factors. Combining the People Related elements of the studies it can be concluded that Collaboration activities and Team/Personal motivation and factors are consistent with other research.

A split has therefore been selected:

People	Collaboration
	Team Integration

Table 3-1 Benefits described in Literature

Type	Theme	By Service Provider (Waters, 2012; Version One, 2013; Layton, 2012; Zolyak, 2013; Ekas, 2012)	By Corporate Research (Begel and Nagappan, 2007)	By Academic Survey (de Cesare et al, 2010; Asnawi, 2012; Vijayasarathy and Turk, 2008)
	Process Management	Early visibility of any quality issues.		
		Early visibility of project issues.		
		Greatly enhanced ability to embrace change.	Short sprints allow flexibility of design	An Agile environment is more flexible being dynamic and constantly adapting.
		The cost of completing each iteration is more predictable.		
		Significantly reduced project risk.		
		Better visibility of project progress.		It is easy for team member to track the progress of their projects under Agile.
		Better product quality.	Improvement in software quality, manifested by having fewer bugs	Improved software quality.
		Always have a working product.		

Methodology			Developers waste less time on tasks they perceive as irrelevant	
				Different variations of Agile should be used for different teams
	Speed of Release	Deliver working software early.	Quick release of working software.	
		Early and regular releases.		Frequent delivery of working software.
				Improved time to market.
People	Collaboration	Much better stakeholder engagement.	Better focus on Customers	Agile Methods focus on customer needs.
		Customers are more satisfied by the deliverables.		
			Improved communication and co-ordination	Business people and developers work together throughout the project - Agile promotes collaboration.
	Collaboration			The most significant benefit of Agile is the involvement of all parties in the development process.
				The daily stand-up meeting was seen

People				to be a significant benefit.
	Team Integration			Companies fully committed to Agile methods have positive perceptions of Agile.
		Higher team morale.		Agile is enjoyable to use.

3.4 Identified Problems with Agile

The articles which specified benefits also describe challenges with Agile. These articles will be reviewed in the same order, starting with the Microsoft Research study (Begel and Nagappan, 2007).

The Microsoft study highlighted difficulties that practitioners had found. The top six difficulties involved in using Agile are:

- Agile does not scale to large projects
- Scrums take too long and are viewed as ways for managers to micromanage
- Management not buying into Agile
- Teams unfamiliar with Agile and there was insufficient training
- Co-ordination with other teams is difficult
- Losing sight of the big picture.

The Microsoft researchers found that some teams had failed to make agile work, with lack of management buy-in being the primary issue. Also there is a view that there is no "one true way to practise Agile Software Development".

The survey by de Cesare (de Cesare et al, 2010) does not specifically questions users about the problems they have found with using Agile, however clear findings have been described indicating that projects fail when:

- Project teams are not motivated.
- Teams do not have the support from management.
- Change control is locked in a traditional project management loop.
- Project progress is still managed in terms of milestones and features and not working software.

From Ani Asnawi's PhD thesis (Asnawi, 2012) a number of challenges with Agile developments have been identified:

- Support from senior management is critical for the success of Agile.
- If team members are not willing to learn new practises, this is a significant challenge to the use of Agile.
- Lack of documentation was not seen as a significant benefit and in some cases for example Government contracts it is an issue. Without specified milestones it is difficult to obtain payment for work which has been completed.
- Where teams are not co-located, Agile is not as effective.
- Teams must be self-organising and empowered to make decisions.

From Vijayasarathy and Turk's (2008) article:

- Organisational Resistance
- Management Apathy
- Inadequate Training

And from case studies (as described in the previous section)

- Many of the benefits of Agile are lost when a telephone is involved in Daily meetings.
- Agile methods are not as effective for large teams. In large projects agile does not promote formal lines of communication.
- Scant documentation can be detrimental especially to large and complex projects.
- It is difficult to make the transition from Waterfall to Agile.
- Increase in technical debt due to developers being over-eager to show their software.

In addition to this, Juyun Cho's PhD thesis (2009) which concentrated on the challenges encountered by Agile teams provides a theoretical analysis of challenges with Scrum and an analysis based on two case studies.

- Where code check-in is not properly managed, there are problems with delivery of the code.
- Teams who are not empowered do not function efficiently.
- New employees should have a warming up period to become familiar with the methodology. If not then team efficiency will suffer.
- Lack of documentation can be detrimental to Agile projects.
- Where customers are not sufficiently involved, projects are likely to fail.
- Where time is not set aside for integration in large projects, dependencies and connections between modules may not be covered, leading to failures in the software.

3.4.1 Common Challenges from Literature

Several challenges appear in existing studies, the most common challenge is that of teams having insufficient support from management. This appears in several guises, that managers themselves do not buy in to Agile and undermine the process, that they do not support teams involved in the difficult process of moving from a Waterfall to an Agile

methodology or that they do not allow Agile teams the confidence to make mistakes.

The problem of scaling Agile to large projects is another recurring challenge. The close collaboration required for Agile appears to become more difficult to maintain when there are large numbers of team members involved in the project.

Most of the articles agree that the transition from Waterfall to Agile is a difficult one. It seems to be difficult for a system based on milestones and sign-offs to make the transition to a flexible Agile approach.

The popularity of off-shore teams has brought its own problems, specifically in Agile projects, where teams that are not co-located are perceived to be less effective.

3.4.2 Breakdown of challenges

In the same way that the benefits have been broken down, challenges will also be grouped into two categories, people-related and methodology-related (Conboy and Coyle, 2011; Dybå and Dingsøyr, 2008, Misra, Kumar and Kumar, 2009). Splitting Methodology features into groups (or themes) does not follow a pattern within published literature. Asnawi (2012) splits process management features into three sections but other literature does not follow the same breakdown.

People factors also do not readily break down into categories, although Asnawi (Asnawi, 2012) split ‘cultural’ and ‘involvement’ factors from the main section this split is not reflected in other literature.

Table 3-2 Challenges described in Literature

Type	By Service Provider (Waters, 2012; Version One, 2013; Layton, 2012; Zolyak, 2013; Ekas, 2012)	By Corporate Research (Begel and Nagappan, 2007)	By Academic Study (de Cesare et al, 2010; Asnawi, 2012; Vijayasarathy and Turk, 2008)
		Agile does not scale to large projects	Agile not as effective for large teams
		Scrum meetings take too long	
		Management do not buy into Agile	Teams do not have support from management.
			Lack of documentation on certain projects, especially large complex ones is a barrier
			Agile not as effective where teams are not co-located
		Lose sight of the big picture	

Methodology			Teams are not empowered to make decisions
			Increase in technical debt
			Change control managed in traditional manner
			Project progress is managed in terms of milestones not working software
People		Teams unfamiliar with Agile and insufficient training	Inadequate training.
		Co-ordination with other teams is difficult	
			Project Teams are not motivated
People			Team members are not willing to learn new practices

3.5 Research Gaps

The majority of articles and theses which have explored the research performed on Agile (Dybå and Dingsøyr, 2008; Lee and Weidong, 2010) have suggested that additional work is required to give Agile a stronger academic justification.

A number of gaps have emerged from the literature review:
Firstly – of primary importance in the context of this thesis, there has been no academic research which has produced data which could be considered to be reliable due to:

- The number of responses within questionnaires has been small.
- Case studies cover only small numbers of different organisations.
- The questions asked have not been comprehensive enough to cover all aspects of Agile.
- Studies are limited to a single Agile methodology.

Secondly - there has been no attempt to connect the features adopted by Agile practitioners and their perception of the benefits and challenges of using an Agile methodology.

Thirdly - there has been limited academic research covering the application of the principals of the Agile Manifesto which is the basis for Agile methodologies and has been as a guide to development using Agile principals. The analysis of benefits and challenges while using Agile methods does provide an insight into the application of the Manifesto, but only so far as might be used as a basis for further research.

Four large-scale surveys have been performed which do have elements of analysis of benefits and challenges while using Agile. However each of the investigations has limitations relating to the collection of data and do not provide a broad-spectrum of practitioners:

- The Microsoft Research survey (Begel and Nagappan, 2007) is limited to staff within a single organisation.
- The perceptions study (de Cesare et al, 2010) is limited to data collected from IT directors, Senior Architects and Managers instead of team members and is more focused on the perceptions of Agile Principals rather than their useage.

- The PhD concerning Agile adoption in Malaysia (Asnawi, 2012) concentrates on Agile adoption and only contains limited information on Agile Benefits and Challenges.
- The Survey of Early Adopters (Vijayasarathy and Turk 2008) ask a very small number of questions relating to challenges and benefits and makes no attempt to provide any link between features and the benefits and challenges.

Case Studies all deal with small groups of users and cover benefits and difficulties relating to single organisation and sometimes projects.

Fourthly - none of the literature compares user satisfaction with Agile to the features that they are employing in their projects, even though there have been many investigations into Agile Adoption. For example the Version One (2014) sponsored ‘State of Agile Development survey’ which is run every year and Ani Aswani’s PhD thesis “Investigating Adoption of and Success Factors for Agile Software Development in Malaysia” (2012).

This thesis will increase the knowledge base regarding user perceptions and will link those perceptions to the features of Agile used.

3.6 Summary

A number of common factors have emerged:

Benefits -

- Improved communication (daily meetings are very useful in achieving this).
- Focus on customer Needs.
- Promotes collaboration.
- Improved product to market time.
- Sort work iterations make it easier to make changes.
- Less irrelevant work is done.
- Improved software quality.

Challenges –

- Agile is not as effective for large projects.
- Where teams are not co-located, Agile is less effective.
- Reducing documentation can be detrimental, more documentation is often required.
- Increase of technical debt.
- When management does not buy into Agile, the process will not work.
- Teams require training and bedding-in before Agile will be effective.
- Easy to lose sight of the big picture.
- Project managers cannot produce the same detail metrics that they can under waterfall.

Effectiveness –

- Agile should be tailored to meet the needs of individual teams.
- Scrum meetings must be carefully managed to ensure they do not take too long.
- Agile can promote too many meetings.
- There is no one way to practise Agile.
- Partial use of Agile is not as effective as fully embracing the methodology.
- Teams must be self-organising and empowered.

Chapter 4. Methodology

4.1 Introduction

Guidance was taken on how data collection and analysis should be performed from recommended texts (Saunders, Lewis and Thornhill, 2009; Oates, 2007). A research plan was devised based on the suggested steps:

1. Identify the objectives of the research
2. Clarify the research area
3. Critical literature review
4. Define a research strategy
5. Negotiate access to data sources
6. Define data sample
7. Collect data
8. Analyse data
9. Produce project report

Sections 1 to 3 have been examined in previous sections of this thesis, sections 4 to 7 will be covered in this chapter.

Because the practice of development teams is materially affected by the perceptions of the actors involved in that development, the ontological assumption is made that a subjective approach will be taken. The epistemology of the research will be interpretive since it is unlikely that hard and fast rules for behaviour of Agile teams will be produced from the analysis. By adopting this approach it has been assumed that understanding of, in this case, software development is based on complex social patterns rather than being based on immovable rules.

A primarily inductive approach will be used when obtaining and analysing data, since there are no assumed conclusions in the research. Inductive reasoning starts at the lowest level of detail and information is build from the bottom level, a pattern is sought, followed by the development of a theory.

4.2 Research Strategy

To achieve the research objectives, mixed research methods were used including a pilot study, a large scale survey and follow up interviews.

A pragmatic research method was employed and mixed research methods were used including a pilot study, a large scale survey and follow up interviews.

Initially discussion with a small group of peers and use of personal experience produced a draft questionnaire, after which a pilot was performed to refine the draft and verify that it was fit for purpose.

The survey form identified the role of each individual, their level of Agile adoption and how they perceived the positive and negative features of Agile. To explain unexpected results and understand more clearly why the expected results appeared as they did interviews were performed to enrich the findings of the survey. The unit of analysis will be individual users.

Based on the research strategy, research was performed as follows:

- The sample was established for the survey.
- A survey instrument was created, based on the research questions which will provide answers to the research questions.
- Initially the questionnaire was piloted to a small group, revised and re-tried with a different small group until a design was produced that met research needs and which was well-received and clearly understood by the pilot groups.
- The final version of the questionnaire was physically handed out where possible.
- A web survey with the same questions was produced to allow access to a wider audience than can be reached by handing out paper forms.
- Through personal and professional contacts, submitted questionnaires to Agile users in three organisations. Then the contacts were requested to ask their associates also to complete a questionnaire, using a snowballing approach to expand the sample size. Contact was made through professional bodies' websites and Agile fora, but no volunteers were found who were willing to distribute forms or promote the survey. Champions were recruited wherever possible and asked to issue and retrieve questionnaires. By adopting this approach a better response rate should be received than the low rate often found in IT surveys (Sivo et al, 2006, Pinsonneault and Kraemer, 1993)
- Contact was made with the moderators of fourteen Agile forums, asking for permission to email a request to forum members.
- Email requests were sent to individuals in Agile forums advertising a link to the web version of the survey.
- Questionnaire data was summarised and analysed for common themes and patterns.

- The analysis will be followed up by interviews with individual users to provide more in-depth insights on how users address the challenges when applying Agile.
- Actions and processes which have been used successfully on Agile/Scrum projects will be identified and collated.

4.3 Access to Data Sources

A convenience sample strategy for collecting data was employed, where the subjects of the survey are selected because of their convenient accessibility and availability to the researcher (Saunders, Lewis and Thornhill, 2009).
The cost and time required to carry out a convenience sample are small and allow the collection of data not possible using probability sampling techniques.

In addition this strategy was preferable because of the reluctance of individuals and organisation to participation in data collections without remuneration. Insufficient funds were available to pay for responses to the survey.

The primary issue with convenience sampling is that the sample is not representative of the entire population and that a convenience sample can lead to the under-representation or over-representation of particular groups within the sample (Davies, 2007). This introduces a sampling bias.

Since the sample is not representative of the entire user base, a convenience sample undermines the ability to make generalisations from the sample for the entire user base. This results in a low external validity of the study, however the aim of the research was not to canvass the entire user population, but to explore commonly found issues and benefits and place those in context (Munn and Drever, 2004).

When the pilot was completed, personal acquaintances were asked to provide completed questionnaires, these individuals were also asked to issue the questionnaires among their workmates and contacts to provide a snowballed group of replies.

Two Agile user groups were joined ‘Agile London’ and ‘Agile Milton Keynes’ and contacts were made at both when attending meetings, the contacts kindly offering to issue questionnaires.
Around eighty hard-copy questionnaires were handed out. Permission was granted for the ‘champions’ in each organisation to hand the questionnaires out.

Application was made to fourteen Agile fora to place a link to the questionnaire on their site. Only two of the moderators responded and only one of those was positive.

Various on-line Agile groups were joined and where these had an open-posting policy, a link to the questionnaire was sent to group members. The majority of the groups were closed and while moderators were contacted for all of these, none responded.

4.4 Data Sample

4.4.1 Who has been approached to provide data

The convenience sample in this case has been obtained through personal contacts and via user groups. It was possible to find a champion within five organisations. The champions were asked to issue hard copy questionnaires to different disciplines within software development teams and Analysts, Project Managers, Developers and Testers all replied from every organisation.

Five companies were approached:
Company One – Provides software to the Education Sector.
400 Employees mostly based in Bedfordshire, but with several offices in other parts of England. Some development is outsourced to Bangalore, India.

Company Two –Industrial, construction and retail group, most supplying material and services to the building sector.
1200 Employees spread all over the UK, whose head office is in Northamptonshire.

Company Three – Produces software for Local Authorities.
40 Employees based in Bedfordshire, but using an outsource company based in Chennai, India.

Company Four – Cloud based reporting and budgeting software used by the Education Sector.
30 Employees based in Derbyshire.

Company Five – International Bank, whose head office is in France.
250 Employees based in London

In addition to this, members of forums and user groups were contacted. Responses in this case were solely those members who were willing to provide a response.

4.5 Choice of data collection method

The requirement stated in the research objective is to Canvas Agile/Scrum professionals for empirical investigation by developing research questions and design instruments for data collection. The data will then be used to assess the level of Agile adoption in the users' organisation and their opinions on the benefits and challenges of using an Agile methodology which will obtain a 'general' view of how Agile is used. Hence as large a data sample as possible was required.

The collection of responses from a large sample in a structured manner lends itself towards use of questionnaire sampling rather than interviews (Saunders, Lewis and Thornhill, 2009). Because it is not possible to interview all Agile practitioners, a sample typical of all possible respondents was decided upon.

The type of information to be obtained was known and understood, the key questions being closed (Munn and Drever, 2004) and data collected should produce a quantitative result allowing the relationship of variables to be established (Newsted, Huff and Munro, 1998). On this basis a questionnaire was chosen as the primary data collection instrument (Saunders, Lewis and Thornhill, 2009).

Some open questions were included in the survey form so that any questions which may have been missed from the issued list of benefits and challenges could be identified.

4.6 Preparing the questionnaire

The questionnaire was developed by following a series of steps. The questionnaire was designed to obtain empirical data about the feature of Agile used by the respondents and how they perceived the benefits and challenges of using an Agile methodology.

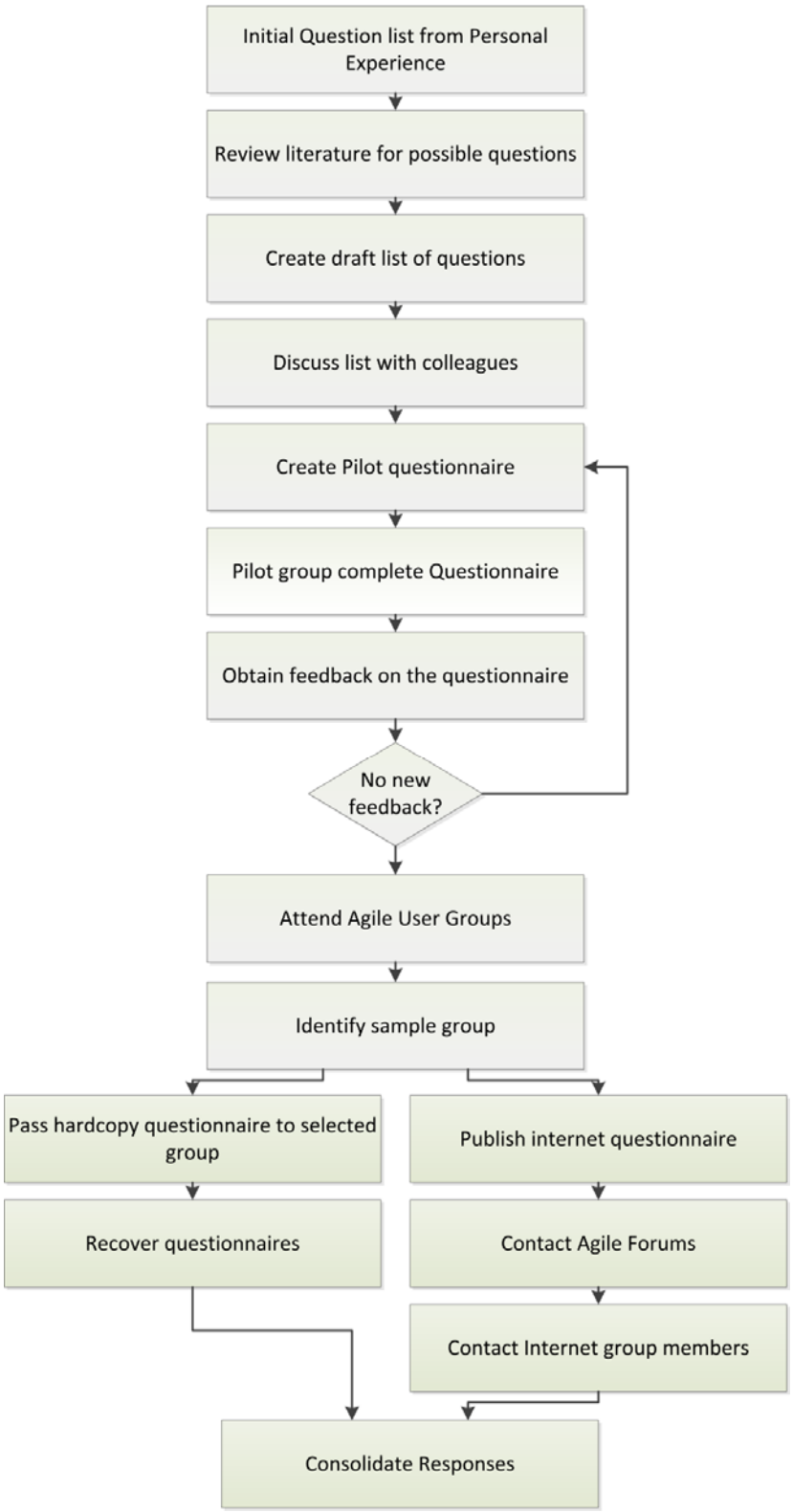

Figure 4-1 Preparing the questionnaire (Oppenheim, 1992)

4.7 Questionnaire Design

A plan was created for the format of the questionnaire which would form five sections

- Details of the responder
- Adopted features of Agile
- User satisfaction with Agile
- Benefits of using Agile
- Challenges found when using Agile

This structure provides information about how users perceive Agile and places it in context with the features of Agile which they use.

Initially questions were formed from personal experience, specifically from the desire to identify what users find useful and frustrating about Agile.

As suggested by Oppenheim (1992) initially a large numbers of questions were generated, which were then reduced. Some questions were poorly phrased, others not relevant though the majority were duplicates of other questions.

The list was then enhanced by those benefits and challenges suggested by Agile academic articles and key books (Anderson, 2010; Leffingwell, 2011; Cohn, 2004).

A mixture of open and closed questions was used in the questionnaire. As many closed questions were included as possible so that a direct comparison could be made between responses, while at the end of the Benefits and Challenges section space was available for respondents to add their own benefits and challenges. A section was also added to allow free text comments regarding what they liked and disliked about Agile.

Different types of scaling were considered for the questions. Thurstone and Guttman scales were considered too cumbersome, but in discussion with my Agile team a Likert scale was decided upon. After trying several different scaling options, a four point scale was chosen for use of features between ‘Always’ and ‘Never’ and for benefits and challenges a three point scale between ‘Not Significant’ and ‘Very Significant’.

The draft questionnaire was discussed with my Agile team who suggested including challenge ‘Stories and Acceptance Criteria are not detailed enough’ and making benefits about collaboration into a single questionnaire line.

Further to this the list of questions was discussed with practitioners from three organisations:

- A senior developer at Company One (Education Software)

- An Agile project manager at Company Five (International Bank)
- An Analysis team leader at Company Three (Local Authority Management Software)

This consultation provides one new Benefit and Challenge and the removal of one Benefit being too similar to an existing question.

The final questionnaire can be seen in Appendix A.

4.8 Pilot survey

Before issuing a questionnaire it is important to test it with a sample group (Oppenheim, 1992) to which purpose the draft form was piloted among a dozen Agile users in my organisation (of which I received eight replies). Suggestions relating to the clarity of some questions were used to improve the questionnaire and the Features, Benefits and Challenges scale was changed to five points, as the pilot group preferred a scale with a defined mid-point.

A second release of the survey was then created and using this, the form was discussed with two experienced practitioners (one in my organisation and one who works in a government body). Again comments regarding the clarity of phrasing of benefits and challenges were absorbed into the survey form.

A third pilot survey was created and this was circulated among a different group of users in two organisations (nine forms issued, six were returned). Feedback again was incorporated into the survey form, the primary change being the addition of a sixth column in the features scale for ‘not sure’ and similarly a sixth column ‘not a benefit’ and ‘not a challenge’ in the Benefits and Challenges section. One new Benefit was added at this stage.

A final pilot was circulated to another six users (five were returned) where this time there were no enhancements requested that had not already been considered and rejected.

At this point the pilot stage of the survey was terminated and a final release version of the survey form was created.

4.9 Released Questionnaire

When the pilot was completed, personal acquaintances were asked to provide completed questionnaires, these individuals were also asked to issue the questionnaires among their workmates and contacts to provide a snowballed group of replies. By using snowballed contacts it was hoped that a better response rate should be received than the low rate often found in IT surveys (Sivo et al, 2006; Pinsonneault and Kraemer, 1993)

4.9.1 The Questionnaire

Around eighty hard-copy questionnaires were handed out. In line with the recommendations in Oppenheim's "*Questionnaire Design, Interviewing and Attitude Measurement"* (1992) and Edwards et al. (2002) the hardcopy questionnaire was printed on cream bonded paper, with titles and headlines in coloured text.
To allow data to be collected from a wider audience an on-line version of the questionnaire was created.

4.9.2 Issuing the Questionnaire

Questionnaires were initially issued through personal contacts. Further personal contacts were sent the questionnaire, though none of these were willing to issue forms among their colleagues. An effort was made when handing out questionnaires to obtain responses from a spread of roles within teams (project manager, business analyst, developer, tester etc.).

By attending Agile user group meetings further contacts were made and a block of completed questionnaires was produced by one company and individual members of the user groups did complete the survey.

To access a wider audience an on-line survey was created with the same questions using the Qualtrics on-line survey tool.
Qualtrics is a flexible, configurable on-line survey tool with basic analytics and the capability to export data in multiple formats. The software has its own simple analysis capability, while supporting data download into Excel or SPSS and has quality control features, such as preventing multiple submissions from a single survey participant and secure 128-bit data encryption (Schvey et. al., 2013).
According to Qualtrics themselves (2013) the software is used by 99 of the top 100 business schools and 1,300 colleges and universities worldwide, including every major university in the United States.

The on-line questionnaire was circulated to personal contacts via email. The contacts were asked to pass details of the questionnaire on to their team members in an attempt to snowball and response.
One source 'Linkedin' allows people in Agile groups who are linked by two degrees of separation to personal contacts on the group to be emailed directly (a feature available on the forum). This yielded a 30% response rate.

A total of 110 completed questionnaires were received. The responses were compiled on Qualtrics, the hardcopy forms were manually loaded into the on-line form and the entry verified by a visual check from another researcher.
An initial result extract was taken from Qualtrics. Then the data was extracted into SPSS for further analysis.

4.10 Post survey interviews

Once questionnaire data has been collated and analysed, additional questions were asked to selected staff to further understand the results of the data analysis (Chen et.al., 2006; Melnik and Maurer, 2006).

Interviews are heuristic and allow the interviewer to gather ideas rather than facts and statistics (Oppenheim, 1992). Interviews were used to explain the quantitative results produced by the questionnaire. Because this study was exploratory in nature, the exact nature of all the data which had to be collected was not known at the time that the questions were created.

The objectives of the interviews are to:

- Identify why unexpected results were found from the questionnaires.
- Be able to explain the questions fully before obtaining a response.
- Receive a more in depth response.

The questions asked in the interviews can be viewed in Appendix B.

Chapter 5. Data Analysis of Agile Questionnaire

The three key sections of the questionnaire (Features used, Benefits and Challenges) consist of closed questions and this analysis will examine the results and relationships between these sections. Some open questions were included which allowed respondents to add their own answers and these will be detailed in the descriptive section.

The analysis will conducted in the following sections:

- Respondent profile
- Adoption of Agile practises
- Descriptive analysis
- Data correlation
- Organisation analysis

The Benefits and challenges have been broken down into two sections, those which relate to People and those relating to the Methodology (Fitzgerald, Hartnett and Conboy, 2006), then further into smaller sections where a logical grouping is present. As with the study by Laanti, Salo and Abrahamsson (2009) no suitable frameworks were found by statistical analysis of the data. The data has therefore been broken down into a logical grouping, based on questions having related themes.

Factors within the data were investigated using factor analysis, but unlike Abbas, Gravell and Wills (2010) no statistical links were discovered. On this basis the following logical grouping was used within data analysis:

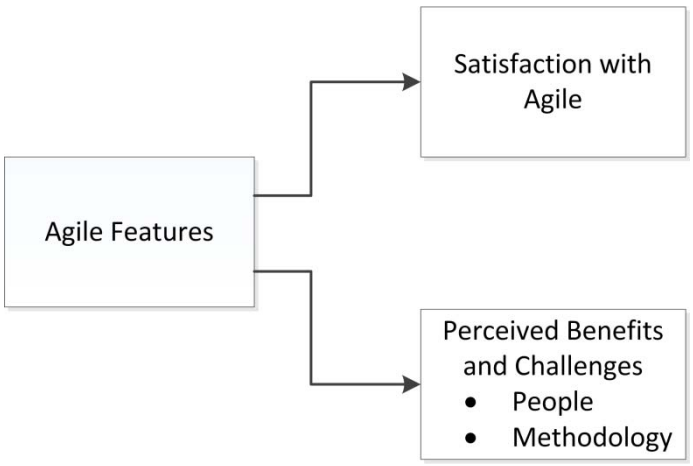

Figure 5-1 Analytical Research Model

5.1 Respondent Profile

To put the results in context, the respondents came from the following types of team member:

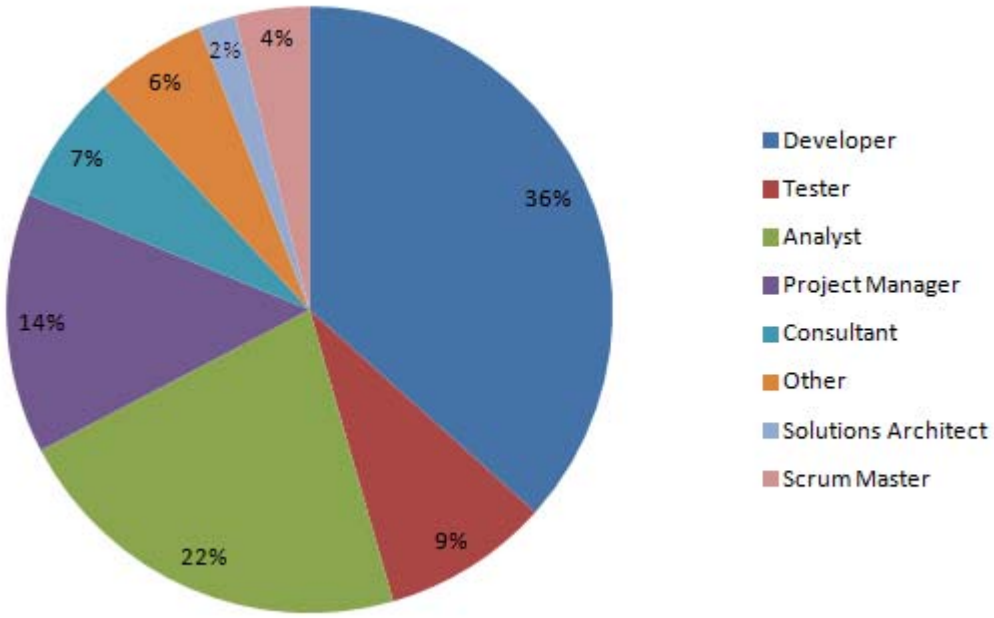

Figure 5-2 Responder Roles

On Average the respondents have been using Agile for 2 years 11 months

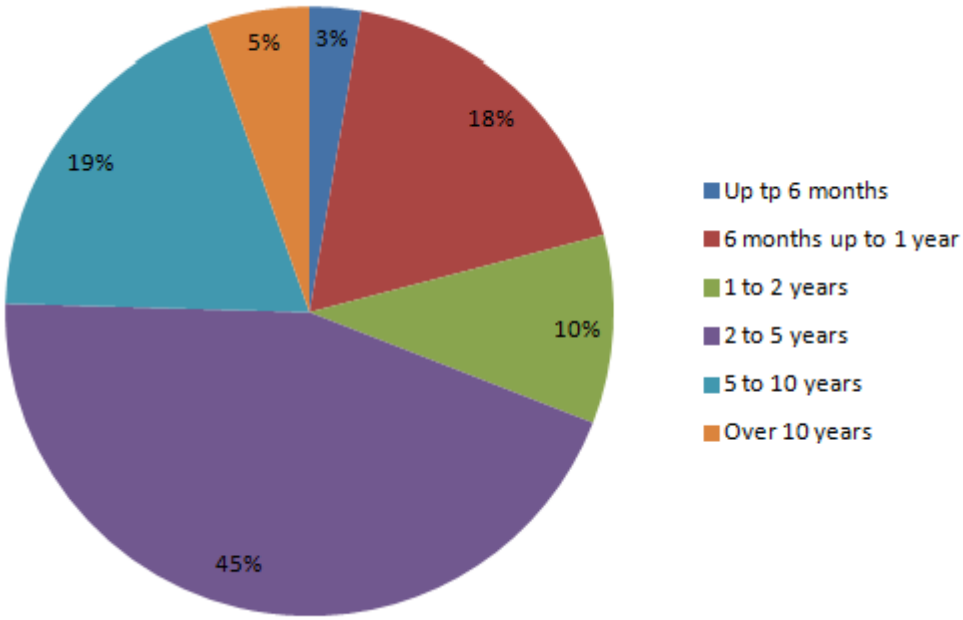

Figure 5-3 Length of time respondents have used Agile

How many staff does your organisation employ?

While the survey was based on a convenience sample and there was no intention to obtain results for a particular spread of organisation size, this question was included to show that there was a spread of responses from different sized organisations.

Table 5-1 Number of staff members in Organisations of respondents

Staff	Response	%
1 - 50	13	11.81
51 - 99	8	7.27
101 - 250	10	9.09
Greater than 250	75	68.18
Not sure	4	3.64
	110	

The respondent profile shows that the questionnaires were completed by practitioners from a wide spread of activities within the team and also within a large spread of different sized organisations.

5.2 Adoption

Because Agile methodologies encourage teams to select the features they wish to use and from observation of Agile teams in action, the features used vary greatly.

Use of Agile features

Table 5-2 Adoption of Agile Features

Always = 5 Never = 1

Feature	Mean	Std Deviation
Stories	4.69	0.763
Daily Scrum Meeting	4.66	0.901
Sprints (short development cycles)	4.61	1.032
Sprint Planning	4.40	1.258
Backlog Queue	4.32	1.347
Retrospectives (post Sprint review sessions)	4.29	1.288
Story Pointing	3.92	1.676
Increments (groups of Sprints)	3.52	1.861
Must/Should/Could/Won't for stories	3.15	1.730
Task Radiator	3.10	1.977
Pair Programming	2.54	1.618

It is possible to deduce from the adoption of features that not only are the components of Agile used to a high degree by the respondents to the questionnaire, but also a wide variation in usage is displayed. The standard deviation in the less commonly used features is approaching two levels.

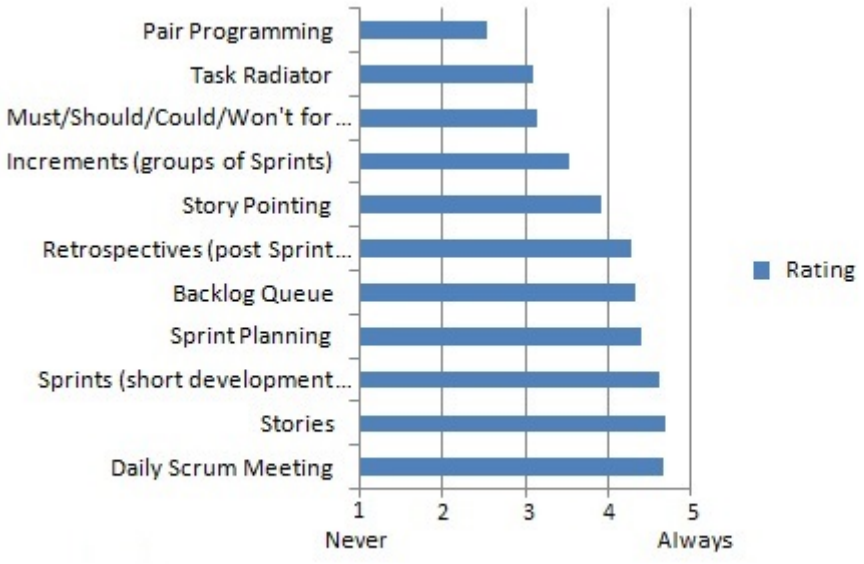

Figure 5-4 Use of Agile Features

There is a very high adoption of daily Scrum meetings, sprints and the use of stories. A much lower use of Story Pointing, one of the key features of Scrum, Kanban and DSDM is shown.

More surprising is the very low take-up of Moscow (Must/Should/Could/Won't) prioritisation of stories which is a key feature of Mike Cohn's "User Stories Applied" one of the most popular manuals for Agile.

Similarly the Task Radiator (sometimes described as a card wall) has a very low take up and this forms a critical part of Scrum and Kanban methodologies (Anderson, 2010) etc.

Because he survey was directed towards Scrum and Kanban it isn't surprising that Pair Programming was the least used feature, as this is a key component of another Agile method, "XP" (extreme programming) and only forms part of Scrum or Kanban when a hybrid methodology has been employed.

Satisfaction with Agile

It appears that Agile is popular amongst a significant majority of the respondents. 90% 0f the participants in the study said they always or usually liked Agile. This is re-enforced by the aggregate scores given to Benefits being far higher than those given to Challenges. This agrees with other studies (Melnik, 2006; Laanti, 2009) suggesting that Agile is generally well-liked.

Do you like using Agile?

Table 5-3 Analysis of users liking Agile

Answer	Response	%
Always	45	40.91
Usually	54	49.09
Occasionally	8	7.27
Rarely	3	2.73
Never	0	0.00
Total	110	

5.3 Benefits and Challenges

5.3.1 Components of the Questions

Benefits and challenges have been broken down into the sections on the right. These will be used to collate data and investigate how Agile is found to be of benefit and difficulty.

The breakdown is different to that shown in previous studies (Chan, 2008; Misra, Kumar and Kumar, 2009) as the factors have emerged from review of the data rather than by reference to a pre-determined model.

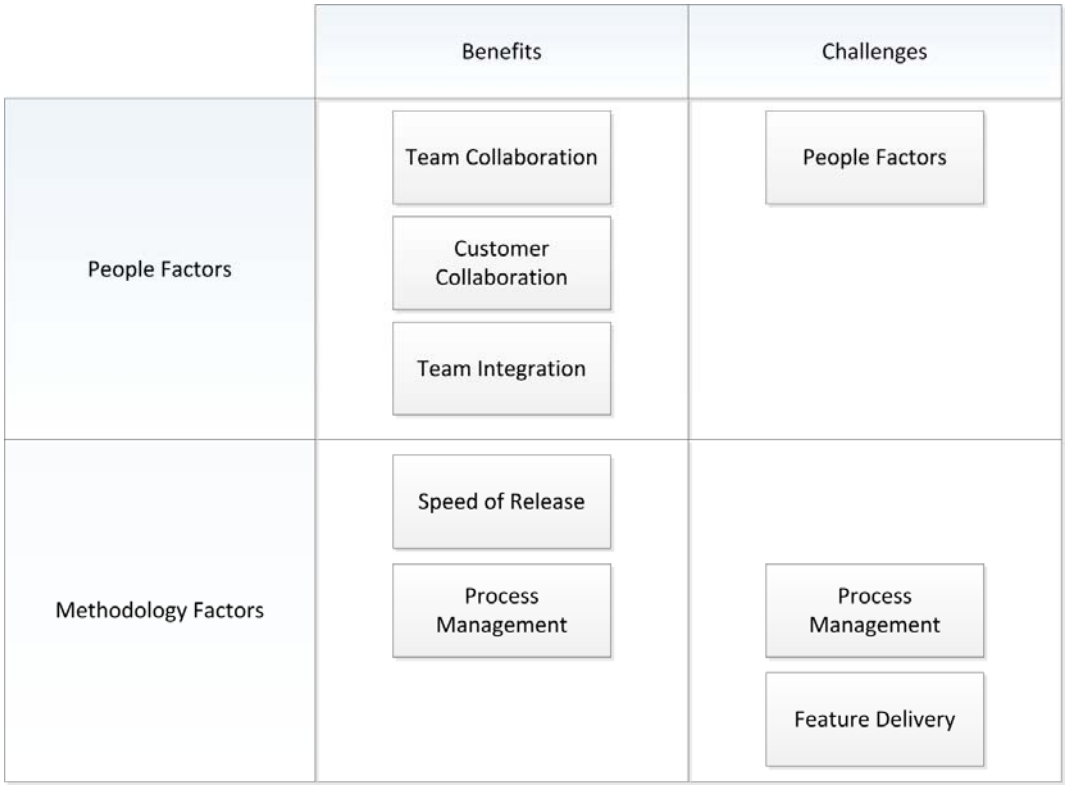

Figure 5-5 Factors broken down by benefits and challenges

5.3.1.1 Methodology factors

Many methodologies are being promoted in the software development world describing themselves as 'Agile'. The methodology selected combined with the way in which it is implemented has a significant effect on the effectiveness of Agile. Many organisations use a hybrid methodology (Barlow, 2011) thereby taking the most useful parts of more than one Agile methodology to provide an appropriate solution for their teams.

Strong management support is required for Agile to be adopted (Pikkarainen, 2012; Cohn, 2003) though self-managing teams will drive Agile methods on their own once

they are fully behind the process.

The questionnaire has been broken down into:

Process Management -

How development is controlled and the implementation of the selected Agile methodologies.

The speed at which software is released –

One of the most celebrated benefits of Agile is that working software is released quickly to customers.

Feature Delivery –

Which features are prioritised and delivered by projects.

5.3.1.2 People factors

The behaviour of team members has to change when teams move to an Agile development method and it is apparent that Agile requires motivated people to succeed. Team and customer collaboration is key to all Agile methods as specified in two of the tenets of the Agile Manifesto (Beck et al, 2001) *Individuals and interactions over processes and tools* and *Customer collaboration over contract negotiation*.
The level of autonomy which Agile teams are given materially effects the effectiveness (Lee, 2010) of the team. In addition the management both of the project and of the organisational features such as the location of the team will also affect the efficacy of Agile (Williams, 2012); face-to-face communication is the preferred option for members of Agile teams.

Because Agile requires team members to be willing and capable of running development themselves, not all members of development teams will be suited to working in an Agile manner. It has been suggested (Nerur et al, 2005; Cohn, 2003) that only above-average development staff should be used in Agile teams.

5.3.2 Benefits of using Agile

People Related

Table 5-4 People related Benefits

5 - most significant 1 - least significant 0 - not significant

Answer	Mean Value	Std Deviation
Team collaboration		
Closer collaboration with team	4.36	0.77
Know exactly what everyone else is doing	3.85	0.91
Predictable communication between team members	3.75	1.13
Everyone on the team has an input to the design	3.56	1.08
Average for team collaboration	3.88	
Customer collaboration		
Working directly with end users	3.53	1.24
Customer involvement	3.53	1.21
Average for customer collaboration	3.53	
Team integration		
Easy to integrate new staff with Agile experience	3.24	1.13
Average value for People Factors	3.69	

Methodology related

Table 5-5 Methodology related Benefits

5 - most significant 1 - least significant 0 - not significant

Answer	Mean Value	Std Deviation
Speed of release		
Being able to see working software quickly	4.23	0.83
Code gets to test earlier due to small deliverable size	4.02	0.95
Shortened feedback loop	4.01	1.00
Bugs are found earlier	3.71	1.05
Average for speed of release	3.99	
Process Management		
Can change the deliverable quickly	3.77	0.99
Developers empowered to manage their own development	3.61	1.17
Reduced Risk	3.52	1.15
Focus on Must have items through MSCW	3.49	1.30
I like the fixed cost and variable features model	3.01	1.24
Less Documentation	2.86	1.13
Average for Process Management	3.38	
Average value for Methodology Factors	3.62	

The average value for the People and Methodology groups of features highlights how both are of equal importance to members of development teams. The scores for the benefits questions all show a high level of significance, demonstrating that the benefits of Agile highlighted by the pilot study are indeed perceived as significant benefits by Agile users.

Several important observations can be made from the data:

- The most important benefit is closer collaboration with the team, which is reinforced by the next three most highly rated questions also relating to communication between team members.
- The speed at which software is available is seen as a highly significant benefit, four of the highest rated Methodology benefits (including the top three) all relate to the benefits of receiving working code quickly.

- While still important, the involvement of customers and end users is not perceived to be as important as the interactions within the development team itself.
- Agile process management is seen as important, but less so that the speed at which working software is available.
- By some margin the production of less documentation is the least important benefit.

5.3.3 Challenges with Agile

Table 5-6 People related challenges

5 - most significant 1 - least significant 0 - not significant

Answer	Mean Value	Std Deviation
Agile doesn't work without competent motivated people	3.40	1.23
Difficulties of using staff inexperienced in Agile methods	2.96	1.39
'No story equals no coding' mentality delays development	2.89	1.33
Less personal accountability as nothing is signed off	2.54	1.30
Average value for People Factors	2.95	

Table 5-7 Methodology related Challenges

5 - most significant 1 - least significant 0 - not significant

Answer	Mean Value	Std Deviation
Process Management		
Hard to estimate work required for a large Agile project	3.68	1.19
Stories and acceptance criteria are not detailed enough	3.38	1.41
Working from very brief requirements	3.09	1.26
No catering for story dependencies	3.06	1.20
Difficult to support after delivery due to lack of documentation	3.05	1.40
Lack of an overview of what is to be done	3.00	1.39
Work is too time driven	2.94	1.32
No history of what has been done	2.88	1.44
Use of stories breaks continuity	2.64	1.41
Average requirement management	3.08	
Feature Delivery		
Always deliver the minimum set of features	2.88	1.37
Sprints are too short to deliver meaningful features	2.70	1.26
Average feature delivery	2.79	
Average value for Methodology Factors	3.03	

Analysis of challenges shows that while significant they are perceived as of far less significance than the benefits.

Several further observations can be made from the data:

- There is little difference between the significance of People or Methodology factors, though poor feature delivery does score as being less significant than any of the other groupings.
- The difficulty involved in estimating the size of Agile projects was the most significant issue.

- The perception of Agile users is that it is impossible to make Agile work without competent and motivated people.
- The format and content of requirements in Agile (using stories) doesn't always provide sufficient information.

5.3.4 Open questions

In addition to the set questions, users were asked to add their own comments about benefits. Each instance of a comment has been noted and where five or more responses made the same comment, they are listed below (the numbers following each show the number of times that comment was made).

Better visibility of progress. **[21]**

Promotes good communication. **[10]**

Similarly for challenges they are as follows (the numbers following each show the number of times that comment was made).

It is difficult to enforce a cultural change. **[5]**

The open questions show

- Better visibility of progress is a key benefit of Agile.
- Improvement in communication is key to Agile reinforcing the high scores shown for both team and customer interaction.
- That Agile represents a significant cultural change which is not embraced by all users.

5.4 Features and their impact on satisfaction

Significance checks were made using the Pearson Bivariate Correlation to check the use of features related to whether the responder liked using Agile.

Table 5-8 Correlation of preference for Agile and Use of features

Feature	Correlation	Do you like using Agile?
Daily Scrum Meeting	Pearson Correlation	.049
Task Radiator	Pearson Correlation	.238*
Backlog Queue	Pearson Correlation	-.008
Stories	Pearson Correlation	.043
Sprint Planning	Pearson Correlation	.067
Sprints	Pearson Correlation	.076
Increments	Pearson Correlation	.013
Retrospectives	Pearson Correlation	.079
Pair Programming	Pearson Correlation	.098
Story Pointing	Pearson Correlation	-.049
Must/Should/Could/Won't	Pearson Correlation	-.094

Only one feature produced a moderate positive correlation which was:

- Users of Task Radiators had a higher satisfaction level using Agile than those who did not use them.

Features by Challenges/Benefits groupings

Using the identified groupings, the following connections have been identified (the table is shown in Appendix C):

Strong Negative Correlation:
Challenges People v Task Radiator
Benefits Customer Collaboration v MSCW

Moderate Negative Correlation:
Challenges Methodology Process Management v Task Radiator
Benefits Customer Collaboration v Stories

Strong Positive Correlation:
Benefits Customer Collaboration v Pair Programming

Moderate Positive Correlation:
Challenges Feature Delivery v MCSW
Benefits Team Collaboration v Pair Programming
Benefits Process Management v Pair Programming

From this we may draw the following conclusions:

- Using the Task Radiator reduces the significance of challenges in an Agile project.
- Pair programming enhances the benefits of People related benefits and Process Management.
- Use of MSCW reduces customer collaboration but improves feature delivery.

5.4.1 Correlation of Features against Benefits and Challenges

Benefits

Features have been checked against individual Benefits and Challenges to determine if there are any links between the responses.

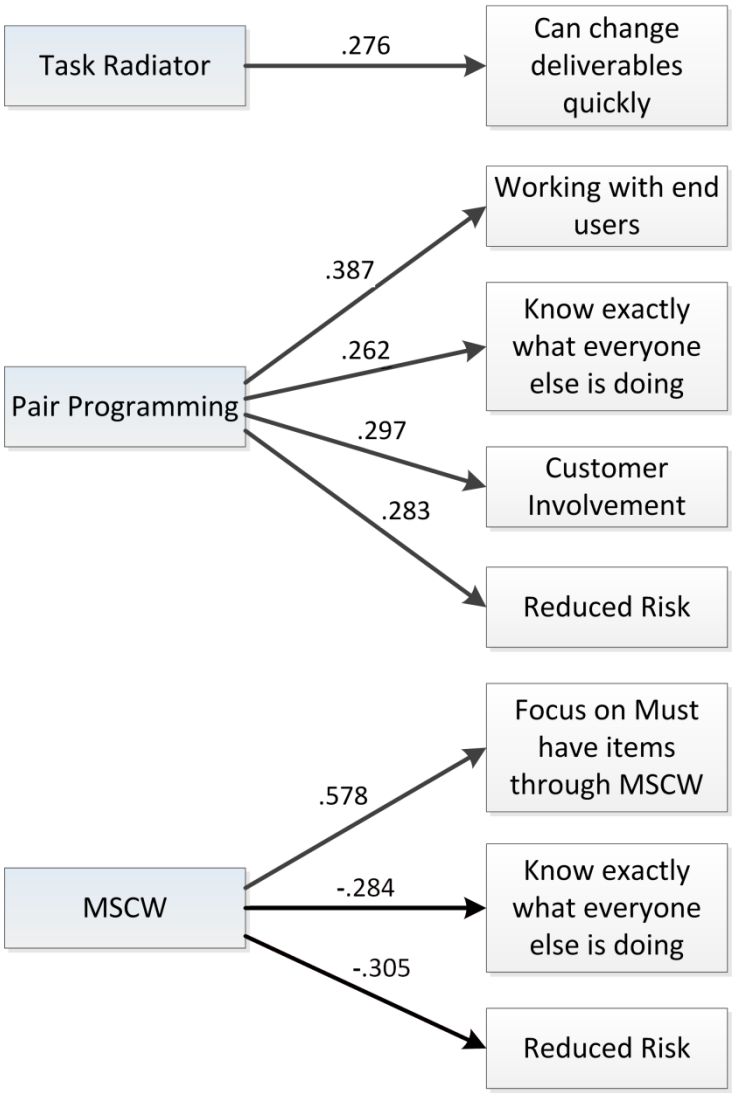

Figure 5-6 Benefits against Features correlation

A strong link is suggested between the use of the task radiator and the speed at which deliverables can be changed, the use of the radiator (sometimes called a 'card wall') is not part of the Kanban (Anderson, 2010) or Scrum (Cohn, 2004) methodologies though its use has become commonplace. This device is intended to make the team's progress immediately visible by tracking the progress of each story. On this basis a link between the use of the radiator and the speed with which deliverables can be changed is expected.

Pair programming requires a significant amount of contact with customers who must form part of the team and must regularly perform tests on small increments of code (Lindstrom and Jefferies, 2004; Sharp and Robinson, 2004). The strong correlation between pair programming and involvement both with customers and other team members is therefore expected.

The use of MSCW and the focus on ‘Must have’ items suggests that the desire to get working software release often does not provide more than the basic set of features for users.

MSCW prioritisation also appears to reduce the awareness of team activities and increases the perceived risk in the project. Further analysis is required to explain why this is the case.

Challenges

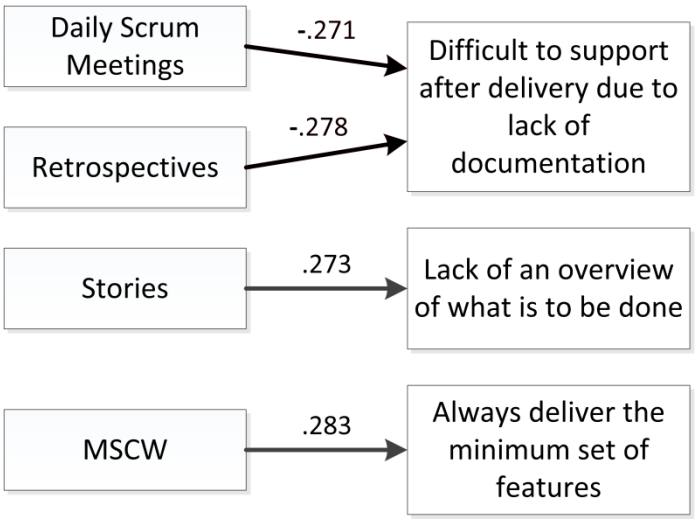

Figure 5-7 Challenges against Features correlation

It appears that the use of daily meetings and retrospectives reduce the difficulties found supporting the system, as powerful communication tools the need for extensive documentation in order to maintain the system after release is mitigated.

The breaking down of requirements into stores does make it more difficult for team members to get an overview of the whole requirement (Augustine, 2005).

The use of MSCW inclines the product owner to deliver only the minimum set of features, since the prioritisation of stories identifies which ‘must’ be completed to produce a viable working deliverable. A system comprising only must have elements will

probably lack the validation and processes which make the software intuitive and easy to use.

5.5 Organisation comparison

Two companies have provided a statistically significant block of responses and these have been investigated separately, in order to identify whether any of their responses are significantly different from the rest of the sample.

Company One:
This organisation has been using Agile for four years, initially selecting DSDM Atern as the methodology of choice, with elements of Scrum. As use of Agile progressed, increasingly teams moved to Scrum and in the last eighteen months, a successful trial was carried on Kanban.
Currently the organisation has a mix of different Agile methods, no two teams using exactly the same process. Roughly half the teams now use Kanban and half Scrum, though there are still a couple of teams using Waterfall.

Company Two:
The organisation has invested heavily in Agile and use Scrum across the organisation while also employing elements of XP. They are heavily involved in the wider Agile community.

Table 5-9 Company comparison

Company One – Education software	**Company Two – Construction and Retail**
Mix of Agile methods, each team selects its own preferred way of using Agile. More teams use Kanban than any other method.	Teams use Scrum/XP.
Uses Task Radiators to display progress.	Do not use Task Radiator
Some teams use Waterfall, some Agile	All teams use Agile
Initial investment in Agile training but very little on-going coaching.	Significant initial and ongoing investment in Agile training and coaching.
Teams controlled by project managers, significant investment in metrics.	No project managers involved in projects, minimal project metrics.

5.5.1 Company results by grouping

Table 5-10 Company breakdown by Challenges and Benefits Groups

Grouping	Company One	Company Two	All
Benefits Team Collaboration	3.7604	3.7000	3.7864
Benefits Customer Collaboration	2.3333	4.0000	3.2409
Benefits Speed of Release	3.8646	3.5250	3.8750
Benefits Procurement Management	2.8333	2.7083	2.9015
Challenges People	1.7188	2.3625	2.1136
Challenges Process Management	2.3194	2.6667	2.2737
Challenges Feature Delivery	1.8958	1.9500	1.8955

There are some significant differences from the average result for the survey

- Company One shows notably lower significance for the benefit of Customer collaboration than Company Two.
- Company One also shows a far lower significance for People related challenges than Company Two

These differences can be explained by the differences in how the companies have chosen to approach Agile.

Company One use Product Managers as proxy customers and requirements are fed to the teams from the product team with little direct involvement between the development teams and actual customers.

Within Company One the respondents to the questionnaire have mostly come from teams with a low staff turnover.
Company Two on the other hand had a much faster transition to Agile, which has been more of a challenge initially for the teams involved.

5.5.2 Challenges by Company in detail

Significant results with a more than 15% difference from the mean. The full table can be found in Appendix C.

Table 5-11 Challenges by Company significantly different from the mean

Question	Company	Company Mean	% of mean	Highlighted in T-test
Lack of overview of what is to be done	n/a	n/a	n/a	Yes
No history of what has been done	Company Two	3.3571	116.77	No
Working from very brief requirements	Company One	3.6667	118.59	No
Work is too time driven	Company One	3.4444	117.20	No
Difficulties of using staff inexperienced in Agile methods	Company One	2.4167	81.56	Yes
Stories and acceptance criteria are not detailed enough	Company Two	4.1579	122.88	Yes

From the analysis we may conclude:

- Difficulties using staff inexperienced in Agile at Company One lower than average.
- Working from too brief requirements and that work is too time driven suggests that Company One are pushing for completed code with unrealistic expectation of how fast that can be achieved.
- Both ‘Stories not being detailed enough’ and ‘insufficient history of what has been done’ score highly at Company Two, which suggests that they have taken the Agile manifesto principal “Working software over comprehensive documentation” too far and that the minimum level of documentation required is not being created.
- It is interesting that Stories not detailed enough and too brief requirements which appear to be very similar questions show such different results between the two organizations.

5.5.3 Benefits by Company in detail

Significant Results with a more than 15% difference from the mean. The full table can be found in Appendix C.

Table 5-12 Benefits by Company significantly different from the mean

Question	Company	Value	% of mean	Highlighted in T-test
Working directly with end users	Company One	2.7500	77.91667	Yes
Customer involvement	Company One	3.0000	84.98584	Yes
Focus on Must have items through MSCW	Company One	4.0417	115.6501	Yes

Company One finds less benefit with user and customer involvement than other organisations.

Company One scores higher on the use of MSCW, but that is explained by Company One having a far higher average score (1.04) for use of MSCW than the average for the whole study of 2.85 on this basis this correlation should not be seen as significant.

5.5.4 Further investigation

Some unexpected results have been produced by the analysis of survey data. These would benefit from further in-depth analysis, to explain the findings.

While most of the highly-scoring benefits, challenges and correlations fall in line with the findings of previous studies or the expectations of service providers, additional investigation can be used to explain the differences.

To this end a group of respondents, who indicated on their survey forms that they were happy to be contacted for further questions were met face-to-face, by phone or by email (based on the opportunity available for each method of contact).

5.5.5 Post Survey Interviews – Data enrichment

Several issues have been discovered by the data analysis which merit further investigation into how Agile practitioners mitigate them.

Nine practitioners who had indicated on their questionnaires that they would be willing to answer further questions were asked to elaborate on the findings, these were selected to include Business Analysts, Testers and Project Managers from several organisations.

Questions were asked in semi-structured interviews, as described in Appendix B. Each interviewee's questionnaire was reviewed before the interview to tailor the questions based on their responses. Interview questions related to the following results of the questionnaire analysis and were posed in order to understand why the unexpected answers were received:

Methodology factors:

1) The difficulty involved in estimating the size of Agile projects was the most significant challenge with using an Agile methodology.
2) The use of MSCW and the focus on Must have items suggests that the desire to get working software released often does not provide more than the basic set of features for users.
3) Within the Benefits sections, one feature scored poorly that of Reduced documentation.
4) Is the use of Scrum meetings and retrospectives useful and do they reduce problems with supporting released software.

People factors:

5) The perception of Agile users is that it is impossible to make Agile work without competent and motivated people.
6) MSCW prioritisation also appears to reduce the awareness of team activities and increases the perceived risk in the project. Further analysis is required to explain why this is the case.

Link between Methodology and People:

7) Task Radiators have appeared as being linked to satisfaction with Agile. Where respondents advised they did not use a radiator how did they keep track of their progress?

Responses to the questions asked:

1.) Estimating projects was the highest scoring challenge to using Agile, those involved in the enrichment were asked whether they agreed with this and how they provided estimates for their Agile projects.

It was agreed that estimation was difficult and one responder commented “The problem with estimating the whole project is that other people want to take your estimate as a promise of delivery.”

For features, ‘T-Shirt’ sizing was used (splitting each feature into Small, Medium, Large, Extra Large etc.) This gives a crude measure for the work involved in each Epic. When relating to sizing in this way, a comment was made “We are getting better at it based on past experience but it is still extremely difficult.”

A total number of points for the whole project can be obtained, based on the previous speed of development of the work involved for that phase of development can be calculated. This is more accurate than the T-Shirt sizing, but the “just-in-time practice of Agile development means that all of the stories for a large project will not be available at the planning stage”.

No-one who replied claimed to have an effective method for estimations, several of the responses stated that they relied heavily on the experience of the team to make initial estimates which tended to be optimistic and all stated that “The initial epic estimates are rarely accurate”. What is to be included in the delivery is also a barrier to making accurate estimates, since deliverables originally thought to be ‘could haves’ are later promoted as ‘must haves’ which throws out the estimates. “... you may receive feedback that the feature was always intended to be included ...”

While it is hard to draw a conclusion from the comments about estimations, it is clear that accurate estimate is impossible at the beginning of a large project, though the estimates improve as the requirements are better understood.

2.) Company One does use MSCW and one of the comments regarding this was that each delivery should not contain only Must have stories. Each response was different on this, one estimated 40% must haves, one 87% and the other 65%. The conclusion which may be drawn from is that the proportion of must have stories varies dramatically between teams.

The responses from Company Two have stated that they do not use MSCW prioritisation to decide the order in which their stories are completed and they were asked how stories were prioritised for delivery. To this they replied that "Stories are prioritised by the business according to their business needs".

3.) Interviewees were advised that 'reduced documentation' had been the lowest scoring benefit of Agile and whether this was correct in their experience.
A couple of people responded that Agile did produce less documentation and that is a good thing but not particularly significant. Another couple reported that Agile did indeed produce less documentation, but that the documentation prepared was now insufficient for development to be properly managed. One responder felt "I think we have gone too far and that there is not enough (documentation)". The consensus of opinion of these four answers is that there is less documentation in an Agile environment than in a Waterfall one.

The five remaining responses commented that there wasn't really a noticeable difference in the amount of documentation produced, but that it was of a different nature. It was also commented that by using Agile tools (TFS and Rally) the process of completing the Agile forms was onerous and outweighed the benefit of generating less documentation.
One responder said "Using Agile there are far more meetings to agree the deliverables" and another stated that "Agile should result in more appropriate documentation, which may be (and often is) less."

4.) When asked whether they felt that the use of Scrum meetings and retrospectives was useful and whether this reduced problems with supporting released software. One commented that "Daily meetings are essential" and another "I don't believe that Scrum meetings help support the software, but they really help communication".

5.) Team members were asked how easy it was to motivate teams to use Agile. All but one response from each company stated that their teams are highly competent and motivated and that the move to Agile was enthusiastically embraced. One person at Company Two said that they did not like agile and were personally resistant to the change, but are having to work under the new system not having any choice. Another at Company One stated that "some people are still not buying in to the concepts and requirements" and that this was largely down to lack of management buy-in.

6.) Use of MSCW prioritisation showed up on the analysis as reducing the awareness of team activities and increasing the perceived risk in the project. None of the enrichment emails or conversations were able to explain why this might be the case, one suggested it was an “anomaly”.

7.) Respondents who stated that they did not use a Task Radiator were asked how they keep track of progress. It appears that the respondents (all from Company Two) use a board with cards that are moved across to denote progress using a process developed by a company called ‘Rally Software’, this appears to be a variation on a standard Task Radiator.

Some questions related to the two companies who provided a large enough sample to be significant. These were targeted at the interviewees who were from Companies One and Two:

8) Company One shows a lower significance for the benefit of Customer collaboration than the general response.
9) Difficulties using staff inexperienced in Agile at Company One is below average.
10) ‘Stories not detailed enough’ and ‘too brief requirements’ which appear to be very similar questions show different results between Company One and Company Two.

Responses to the company specific questions:

8.) The responses from Company One had a number of explanations regarding why the benefits they found from customer collaboration might be lower than average. Two of the responses felt that with a very large customer base it was difficult to find a consensus of customer opinion “(customers) may have different or even conflicting views of how to approach things, or what is important to them that means it is sometimes makes it difficult to have a consensus of approach” and because of this there was a great reliance on the Product Manager acting as a proxy customer.

The other responder suggested that “We have traditionally seen customers as the bill payers, rather than the end users” and that because the majority of customers were not seen as expert users, their views were often not seen to be important.

9.) Within Company One the enrichment questions were aimed at members of teams where the majority of the questionnaires had been answered, where the respondents advised that there had been a very low turnover of staff in the last couple of teams (this is also the case in my team).

10.)The strange result which appeared during the statistical analysis, that “Stories not being detailed enough” and “working from too brief requirements” scored very differently from each other, Company One scored the first as important and the second as not significant, Company Two scored the second as significant and the first as not significant.

Not only are the two similar questions scored differently, but also the significance was different for the two companies. On posing the question why this might be, the respondents could not explain this difference and perceived the questions as being the same. One responder did comment that “the stories themselves have enough information about the circumstances and objective at a higher level but don’t have enough requirements to guide the developers (sic)”

5.5.5.1 Summary of Data Enrichment

Nine practitioners were interviewed and each was asked questions tailored to their questionnaire responses. The survey responses which produced unexpected results were the focus of the interviews.

Estimation of effort required to complete projects is agreed to be difficult by all respondents, a variety of methods are used to create estimates. Because requirements are not as well defined in an Agile environment as they are in Waterfall, estimates will be made on less clearly defined deliverables and therefore less accurate.

The claim that Agile requires reduced documentation is generally felt to be correct, though this is not always believed to be of benefit. Sometimes there is no reduction in recording the project because a tool is used to replace project documentation, using the tool requires as much effort as producing the project artefacts it replaces. On other occasions, documentation is reduced which leaves insufficient guidance for project management and development.

Where a task radiator is not used (in this case at Company Two) a software tool is employed instead which performs the same function.

Company One shows a low score for the benefit of customer collaboration, which can be explained by the company culture of using Product Managers to define requirements rather than the development teams engaging with actual customers.

5.6 Comparison with previous studies

To further highlight the new insights of this study and demonstrate the contributions, this section attempts to compare the benefits and challenges claimed by Service Providers, Microsoft Research and Academic studies in the literature with the results of this study.

In this section, where the term pilot study has been used, it relates to both the initial series of interviews and discussion with colleagues and the four iterations of questionnaire pilot.

Many of the claims discovered during literary review were removed at the pilot stage when they were felt not to be a significant benefit or challenge. The two tables in this section show the comparison between the literature review and the results of data analysis.

Where the benefits and challenges were not included in the pilot study, or have not been highlighted by open questions or data enrichment, they have been excluded from the tables.

The highest scoring benefits which were included in the questionnaire were highlighted by service providers, but many benefits which were included especially those around team collaboration do not feature as significant within existing literature. This includes the highest scoring benefit ‘Closer collaboration with the team’.

Benefits not selected do not follow a clear pattern. Improved quality was discussed during the pilot phase and was not considered to be a significant improvement found when moving to an Agile methodology. More commercial benefits like ‘Improved time to market’ and ‘early releases’ were not included because the focus of this research was Agile team members rather than senior management who have a more sales-orientated focus and were strongly represented in other studies (de Cesare et al, 2010; Vijayasarathy and Turk 2008).

Employment of daily scrum meetings was suggested to be of benefit by literary review, though the meetings were sometimes too long. This was not requested to be included in the survey form at the pilot stage, or identified in open questions or follow-up interview even though scrum meetings scored as having the highest level of adoption of all Agile features.

Because customers were not canvassed, it is impossible to comment on their satisfaction with Agile. Similarly a large enough sample of companies was not available from the returned questionnaires to be able to judge how much fully committing to Agile improves the perception of it as a way to develop software. This has been considered by Ani Asnawi’s PhD thesis (2012) though there is not yet a large enough sample to make a generalised statement.

Many challenges (similarly to the benefits) have not featured as important in previous studies, these having been requested during the pilot. This includes the highest rated challenge showing the difficulty of estimating large Agile projects.

Challenges not selected by the pilot study generally relate to project management and the control of projects, this reflects the difference in emphasis between previous investigations of Agile and this study.

Table 5-13 Survey Benefits against literary review

Ratings: 5 – most significant 1- least significant

Benefits reported in the literature	Service	Corporate	Academic	Result of this study	Rating in this study
Early visibility of any quality issues.	X			Seen as a very significant benefit.	4.02
Early visibility of project issues.	X			Seen as a very significant benefit.	4.01
Short sprints allow flexibility of design	X	X	X	Seen as a significant benefit	3.77
Significantly reduced project risk.	X			Seen as a significant benefit	3.52
Developers waste less time on tasks they perceive as irrelevant			X	Seen as a significant benefit	3.61
Quick release of working software.	X	X		Seen as the most significant benefit in methodology.	4.23
Much better stakeholder engagement.	X	X	X	Phrased as 'customer' not 'stakeholder' but seen as significant.	3.53
Customers are more satisfied by the deliverables.	X	X	X	Customers were not canvassed .	
Improved communication and co-ordination		X	X	Seen as a significant benefit.	3.75

Involvement of all parties in the development process.			X	Seen as a significant benefit.	3.56
Companies fully committed to Agile methods have positive perceptions of Agile.			X	Not the focus of this research, would require further investigation.	
Higher team morale.	X		X	Not the focus of this research, but 90% of responders say they like Agile.	

Table 5-14 Survey Challenges against literary review

Ratings: 5 – most significant 1- least significant

Challenges reported in the literature	Service	Corporate	Academic	Result of this study	Rating in this study
Management do not buy into Agile		X	X	Post survey interview showed this to be a problem.	
Lack of documentation on certain projects, especially large complex ones is a barrier		X		Shown as a significant challenge	3.05
Lose sight of the big picture			X	Seen as a significant challenge.	3.00
Teams are not empowered to make decisions		X		Not highlighted as a Challenge during pilot.	
Inadequate training.		X	X	Phrased as the difficulty of using staff inexperienced in agile.	2.96
Project Teams are not motivated		X		Seen as the most significant people challenge.	3.40
Team members are not willing to learn new practices		X		Highlighted as an issue in the response to open questions.	

5.7 Summary

The data has displayed connections in several areas, some of which can be expected by previous studies and Agile service providers claims for the process:

- The speed with which software is delivered is seen as more important than improved process management.
- The most important benefit with Agile is closer collaboration with the other members of the development team.
- It is impossible to make Agile work without competent motivated people.
- Agile requires a cultural change which is not embraced by all users.
- Pair Programming enhances the benefits of Customer Collaboration and Process Management.
- Use of MSCW prioritisation improves feature delivery.

Some of the findings are not expected:

- Users of Task Radiator have a higher level of satisfaction with Agile that those who do not use it.
- There is no difference between the importance of People Challenges and Methodology challenges.
- The difficulty involved in estimating Agile projects was the most significant challenge users experience with the method.
- The level of feature adoption has no significant effect on the perceived benefits or challenges with Agile.
- Customer involvement is not as important as team involvement, though because actual customers are not always available and a proxy such as a Product Manager is used, further investigation is required to validate this.

Some of the findings go against the claims made for Agile:

- That the format and content of stories does not always provide sufficient information for development and post implementation support.
- Use of MSCW reduces customer collaboration.

Chapter 6. Conclusion

6.1 Overview

Many claims have been made about Agile methodologies since their introduction, but very little academic research has been carried out to understand the views of Agile users on its benefits and challenges. The purpose of performing this research was to investigate how Agile practitioners use methodologies and how they perceive its benefits and challenges.

By obtaining users' views about Agile from a sample of over one hundred practitioners, it has been possible in this study to examine the claims made for Agile by its adherents. A systematic literature review of the analysis of benefits and issues with Agile concluded that:

- There has been no academic research based on a large number of responses to identify the users' perceptions of Agile, though this has been touched on by several articles.
- There has been no attempt to connect the features adopted by Agile practitioners and their perception of the benefits and challenges of using an Agile methodology.
- Four large-scale surveys have been performed which do have elements of analysis of benefits and challenges while using Agile. However each of the investigations has limitations relating to the collection of data and do not provide a broad-spectrum of practitioners.

A pragmatic research perspective was employed and mixed research methods were used including a pilot study, a large scale survey and follow up interviews.

Initially, discussion with a small group of peers and use of personal experience produced a draft questionnaire, after which a pilot was performed to refine the draft and verify that it was fit for purpose.

The survey form identified the role of each individual, their level of Agile adoption and how they perceived the positive and negative features of Agile. To explain unexpected results and understand more clearly why the expected results appeared as they did, interviews were performed to enrich the findings of the survey.

The company comparison did display some differences but they can be explained by the differences in how the companies have chosen to approach Agile and there were no notable insights into how Agile could be improved.

6.2 Objectives

The objectives of the study have been described in section 1.2, the extent to which they have been met is as follows:

Table 6-1 How objectives have been met

Objectives.	How the objectives have been met.
Understand the current issues surrounding the Agile application through literature review.	• A comprehensive review of academic and practitioner literature has been completed.
Understand how Agile users are using Agile features, their level of satisfaction and perceived benefits and challenges by collecting empirical evidence from Agile practitioners in different organisations.	• Agile/Scrum professionals were approached for empirical investigation. • Research questions were developed. • A survey questionnaire for data collection was produced, piloted and revised. • 110 survey forms were collected. • Data were analysed and the key findings were summarised, the conclusions are described in section 6.2 below.
Identify areas of Agile that are seen to be problematic and understand how users are addressing commonly found issues.	• Appropriate users were selected for interview. • Nine interviews were conducted to investigate how issues were addressed.
Raise awareness and disseminate the findings for the benefit of the wider Agile community.	• Appropriate channels for dissemination were identified and approached. • A summary document will be presented to Agile user groups, academic researchers, described in section 6.6 below.

In general, the aims and objectives of this project have been achieved to the extent that time, resources and experience allowed.

6.3 Summary of findings from data analysis

Many of the results of data analysis were suggested by previous studies, but fewer of the claims of service providers were given justification.

The most important benefit of using Agile was closer collaboration with the other members of the development team and the most significant challenge was the estimation of large projects. Because of the way that Agile breaks requirements into small pieces, each of which is estimated just in time for the next development sprint, Agile teams are conditioned not to think in terms of hours of work.

A striking further insight is that team interactions were seen as the most significant benefit of Agile, scoring far higher than interaction with customers.

The respondents also believe it is impossible to make Agile work without competent motivate people and that use of Agile requires a cultural change which is not embraced by all users.
It was also felt that strong support for Agile from Management is essential for successful use of Agile methodologies.

Table 6-2 Claimed Benefits against findings

Claimed Benefit of Agile	Result of data analysis
Early visibility of project issues, reduced risk.	Supported by the data.
The cost of completing each iteration is more predictable.	Not suggested as a benefit.
Always have a working product.	Not suggested as a benefit.
Customers are more satisfied by the deliverables.	Customers not canvassed.
Developers waste less time on tasks they perceive as irrelevant	Not suggested as a benefit.
Enhanced ability to embrace change.	Supported by the data.
Better visibility of project progress.	Supported by the data.
Better product quality.	Not suggested as a benefit.
Frequent, early delivery of working software.	Supported by the data.
Improved time to market.	Not suggested as a benefit.
Improved collaboration and co-ordination.	Supported by the data.
All parties are involved in the development process.	Supported by the data.
The daily stand-up meeting is a significant benefit.	Supported by the data.
Companies fully committed to Agile methods have positive perceptions of Agile.	Insufficient companies canvassed.
Agile is enjoyable to use, with higher team morale.	Supported by the data.

As shown in table 6.2 some of the claims made for Agile are supported by the data collected in this study, some were not tested as an insufficient number of companies and no customers were canvassed. Several of the claims were not suggested by the initial collection of benefits and were not raised by the pilot or by the responders to the questionnaire as benefits.

Significant findings that were not predicted by previous research include:

- Users of Task Radiator have a higher level of satisfaction with Agile that those who do not use it.
- There is no difference between the importance of People Challenges and Methodology challenges.
- The difficulty involved in estimating Agile projects was the most significant challenge users experience with the method.
- The level of feature adoption has no significant effect on the perceived benefits or challenges with Agile.

- Customer involvement is not as important as team involvement, though because actual customers are not always available and a proxy such as a Product Manager is used, further investigation is required to validate this.
- Users believe that Agile works extremely well for small teams but does not scale well for large complex projects.
- That the format and content of stories doesn't always provide sufficient information for development and post implementation support.
- Use of MSCW reduces customer collaboration.

Adoption of Agile features is inconsistent, even within organisations. Daily meetings and the use of incremental builds are the most commonly used features.
The use of the task radiator has been shown to correlate with Agile users' satisfaction, while the absence of other features which effect satisfaction is also significant.

6.4 Contributions of this study

The primary contributions of this research are:

- To provide knowledge and information about how Agile practitioners perceive the benefits and challenges of their methodologies. The literary review showed a paucity of studies in this area with no study specifically investigating benefits and challenges by survey.
- To compare the claims made for Agile with the empirical results so as to provide evidence based justifications for Agile applications from Agile practitioners' perspective.

The insights gained can be used as a basis for further research and used in practice as a guide to where the most significant benefits and issues may be found when using Agile.

Although benefits and challenges highlighted by the literature review broadly agree with those found in this investigation, this research re-enforces many previous findings, expands upon them and has discovered some elements not previously recorded.

6.5 Implications for research and practise

There is a need for further academic study of how users perceive Agile and the application of this to practise, but this study suggests that:

- Agile teams should use a task radiator.
- Improved co-operation within teams can be expected when development is performed in an Agile way.
- It will be difficult to estimate the work and timescale of large Agile projects.

An in-depth study of practitioner enjoyment and how the like using Agile correlated against the features and Agile methods being used could be developed into a selection method to match the most appropriate features for Agile teams.

6.6 Disseminating the findings to the Agile community

In order to make the research available to Agile practitioners and academics, a summary of the research findings has been created and will be presented to the Academic Agile conference at Lancaster University in July 2015.

All of the respondents to the survey who asked to be contacted with the results will have the summary document sent to them.

The research summary will be submitted to VersionOne who collect information for a large-scale Agile survey every year and are well-known in the Agile world. The VersionOne moderator has expressed interest including the research in the news section of their website.

The summary will be published on the Agile Business Analysis, Test and Development forums on Linkedin and also on the Scrum Alliance forum page.

6.7 Limitations and Further Research

Other possible extensions of the project are discussed together with ideas for applications on totally different contexts.

6.7.1 Improvements and extensions on the current work

The research within this and other studies would be enhanced by posing similar questions to a larger group of subjects and applying a more random sampling mechanism to identify candidates for the survey. By collecting more data, a more generalised view of Agile could be obtained.

Three additional criteria which appeared frequently in the user defined benefits/challenges section where users were able to add their own items should be included in the survey form:

Benefits:	Better visibility of progress.
	Promotes good communication.
Challenges:	It is difficult to enforce a cultural change.

A similar questionnaire could be issued on a regular basis, using the data from this it would be possible to see how practitioner views of Agile change over time.

6.7.2 Limitations

The validity of data produced by this study was limited by the relatively small sample size of 110 responses, so conclusions may not be generalised.
Only two of the organisations contacted were able to generate more than 20 replies, only a small sample of the total number of employees in the organisations. The convenience sampling which provided the responders to the survey is obviously biased by only those practitioners who could easily be convinced to take the survey responding.

Finances and time were limited in the research, being able to offer financial incentives for example would have increased the number of questionnaires that would have been collected (Robin and Nash, 1973)

Data collection and analysis may also be influenced by skills and experience of the researcher.

Research by survey has the benefit of being able to collect information from a large number of people in a predictable format for a modest cost. The validity of the questionnaire has proven to be high, since the data collected does reflect the original purpose, however the data collected is self-reported by the responder and the reliability of the questionnaire is difficult to estimate.

Reliability in this case, being the accuracy and dependability with which the survey measures the attribute it is designed to measure. There is always a degree of uncertainty because there is no way to tell how truthful a respondent is being, or how much thought they have put into their answers. In addition to this the researcher has created their instrument based on their own perceptions and assumptions. This study presupposes a degree of knowledge and understanding of Agile methods in its respondents. Steps were taken to ameliorate any unreliability by performing several stages of pilot survey and post-survey interviews to validate the data.

References

ABBAS, N. 2009. Software quality and governance in agile software development. University of Southampton.

ABBAS, N., GRAVELL, A. M. & WILLS, G. B. Using factor analysis to generate clusters of agile practices (a guide for agile process improvement). Agile Conference (AGILE), 2010, 2010. IEEE, 11-20.

ABRAHAMSSON, P., SALO, O., RONKAINEN, J. & WARSTA, J. 2002. Agile software development methods: Review and analysis, Oulu, Finland, VTT Finland.

AL-BAIK, O. & MILLER, J. 2014. The kanban approach, between agility and leanness: a systematic review. Empirical Software Engineering. 1-37.

ANDERSON, D. J. 2010. Kanban, Blue Hole Press.

ASNAWI, A. L. 2012. Investigating adoption of and success factors for agile software development in Malaysia. University of Southampton.

ASNAWI, A. L., GRAVELL, A. M. & WILLS, G. B. Emergence of agile methods: perceptions from software practitioners in Malaysia. AGILE India (AGILE INDIA), 2012, 2012. IEEE, 30-39.

AUGUSTINE, S., PAYNE, B., SENCINDIVER, F. & WOODCOCK, S. 2005. Agile project management: steering from the edges. Communications of the ACM, 48, 85-89.

BARLOW, J. B., GIBONEY, J. S., KEITH, M. J., WILSON, D. W., SCHUETZLER, R. M., LOWRY, P. B. & VANCE, A. 2011. Overview and guidance on agile development in large organizations. Communications of the Association for Information Systems, 29, 25-44.

BATRA, D., XIA, W., VANDERMEER, D. & DUTTA, K. 2010. Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry. Communications of the Association for Information Systems, 27, 21.

BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R., MELLOR, S., SCHWABER, K., SUTHERLAND, J. AND THOMAS, D. 2001. Agile manifesto. Online at http://www. agilemanifesto. org.

BEGEL, A. & NAGAPPAN, N. Usage and perceptions of agile software development in an industrial context: An exploratory study. Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on, 2007. IEEE, 255-264.

BENINGTON, H. D. Production of large computer programs. ICSE, 1987. 299-310.

BEYNON-DAVIES, P., CARNE, C., MACKAY, H. & TUDHOPE, D. 1999. Rapid application development (RAD): an empirical review. European Journal of Information Systems, 8, 211-223.

BEYNON-DAVIES, P., MACKAY, H. & TUDHOPE, D. 2000. 'It's lots of bits of paper and ticks and post-it notes and things...': a case study of a rapid application development project. Information Systems Journal, 10, 195-216.

BOEM, B. 1995. Spiral model of software development and enhancement. IEEE Engineering Management Review, 23(4), 69-81.

BOOCH, G. 1986. Object-oriented development, IEEE Transactions on Software Engineering, SE-12 (2), 211 – 22.

BROOKS, R. E. 1980. Studying programmer behavior experimentally: the problems of proper methodology. Communications of the ACM, 23, 207-213.

CHAN, F. K. & THONG, J. Y. 2009. Acceptance of agile methodologies: A critical review and conceptual framework. Decision Support Systems, 46, 803-814.

CHAUHAM, A. 2003. Rice University - Lecture 2: Software Lifecycle and Team Programming
[Online] Available from:
http://www.clear.rice.edu/comp314/lec/week2/sw-lifecycle-notes.html [Accessed: 23rd March 2014]

CHEN, S., DUAN, Y., EDWARDS, J. S. & LEHANEY, B. 2006. Toward understanding inter-organizational knowledge transfer needs in SMEs: insight from a UK investigation. Journal of knowledge management, 10, 6-23.

CHO, J. 2009. A Hybrid Software Development Method For Large-Scale Projects: Rational Unified Process With Scrum. Journal of Issues in Information Systems, 5, 340-348.

COCKBURN, A. 2004. Crystal clear: a human-powered methodology for small teams, Pearson Education.

COCKBURN, A. & WILLIAMS, L. 2003. Agile software development: It's about feedback and change. Computer, 36, 0039-43.

COHN, M. 2004. User stories applied: For agile software development, Addison-Wesley Professional.

COHN, M. & FORD, D. 2003. Introducing an agile process to an organization. Computer, 36, 74-78.

CONBOY, K. 2009. Agility from first principles: reconstructing the concept of agility in information systems development. Information Systems Research, 20, 329-354.

CONBOY, K., COYLE, S., WANG, X. & PIKKARAINEN, M. 2011. People over process: Key challenges in agile development. Ieee Software, 28, 48-57.

DAVIES, M. 2007. Doing a Successful Research Project. Palgrave MacMillan.

DE CESARE, S., LYCETT, M., MACREDIE, R. D., PATEL, C. & PAUL, R. 2010. Examining perceptions of agility in software development practice. Communications of the ACM, 53, 126-130.

DYBÅ, T. & DINGSØYR, T. 2008. Empirical studies of agile software development: A systematic review. Information and software technology, 50, 833-859.

EDWARDS, P., ROBERTS, I., CLARKE, M., DIGUISEPPI, C., PRATAP, S., WENTZ, R. & KWAN, I. 2002. Increasing response rates to postal questionnaires: systematic review. Bmj, 324, 1183.

EKAS, L. 2012 Five benefits of agile practices in software-intensive systems development
[Online] Available from: https://www.ibm.com/developerworks/community/blogs/invisiblethread/entry/five_benefits_of_agile_practices?lang=en [Accessed: 8 March 2013]

ERICKSON, J., LYYTINEN, K. & SIAU, K. 2005. Agile modeling, agile software development and extreme programming: the state of research. Journal of Database Management (JDM), 16, 88-100.

FITZGERALD, B. 1997. The Use of Systems Development Methodologies in Practise: A Field Study. The Information Systems Journal 7, 201-212.

FITZGERALD, B., HARTNETT, G. & CONBOY, K. 2006. Customising agile methods to software practices at Intel Shannon. European Journal of Information Systems, 15, 200-213.

GOTH, G. 2009. Agile tool market growing with the philosophy. Software, IEEE, 26, 88-91.

GUNTAMUKKALA, V., WEN, H., TARN, J. 2006 An empirical study of selecting software development life cycle models. Human Systems Management. 25 (4), 265-278

HIRANABE, K. 2007. Visualising Agile Projects with Kanban Boards.
[Online] Available from: http://www.drdobbs.com/architecture-and-design/visualizing-agile-projects-with-kanban-b/201807863 [Accessed: 20th October 2014]

HOLTSNIDER, B.,WHEELER, T., STRAGAND, G., GEE, J.2010.Agile Development and Business Goals, Morgan Kaufman Publishing

HONG, W., THONG, J. Y., CHASALOW, L. C. & DHILLON, G. 2011. User acceptance of agile information systems: a model and empirical test. Journal of Management Information Systems, 28, 235-272.

HOWARD, A. 2002. Rapid Application Development: rough and dirty or value-for-money engineering? Communications of the ACM, 45, 27-29.

HURTADO, J. 2013. Open Kanban - An Open Source, Ultra Light, Agile & Lean Method. [Online] Available from: http://agilelion.com/ [Accessed: 7 June 2015]

KHAN, A., QURESHI, M., KHAN, U. 2011. A Comprehensive Study of Commonly Practiced Heavy & Light Weight Software Methodologies. International Journal of Computer Science and Issues, Vol. 8/4, 441-450

LAANTI, M., SALO, O. & ABRAHAMSSON, P. 2011. Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. Information and Software Technology, 53, 276-290.

LADAS, C. 2009. Scrumban-essays on Kanban systems for lean software development, Seattle, WA U.S.A, Modus Cooperandi Press.

LAYTON, M. C. 2012. Agile project management for dummies, John Wiley & Sons.

LEE, G. & XIA, W. 2010. Toward agile: an integrated analysis of quantitative and qualitative field data. Management Information Systems Quarterly, 34, 7.

LEFFINGWELL, D. 2010. Agile software requirements: lean requirements practices for teams, programs and the enterprise, Addison-Wesley Professional.

LINDSTROM, L. & JEFFRIES, R. 2004. Extreme programming and agile software development methodologies. Information Systems Management, 21, 41-52.

MARTIN, J. 1991. Rapid application development, Macmillan Publishing Co., Inc.

MCAVOY, J. & BUTLER, T. 2009. The role of project management in ineffective decision making within Agile software development projects. European Journal of Information Systems, 18, 372-383.

MELLOR, S. J. 2005. Editor's Introduction: Adapting Agile Approaches to Your Project Needs. Ieee Software, 22, 0017-20.

MELNIK, G. & MAURER, F. 2006. Comparative analysis of job satisfaction in agile and non-agile software development teams. Extreme Programming and Agile Processes in Software Engineering. Springer.

MESSENGER, S. 2014. The future is worth the risk. The Independent, April 2014.

MISRA, S. C., KUMAR, V. & KUMAR, U. 2009. Identifying some important success factors in adopting agile software development practices. Journal of Systems and Software, 82, 1869-1890.

MNKANDLA, E. 2008. A Selection Framework For Agile Methodology Practices: A Family of Methodologies Approach. Faculty of Engineering and the Built Environment, University of The Witwatersrand.

MOE, N. B., AURUM, A. & DYBÅ, T. 2011. Challenges of shared decision-making: A multiple case study of agile software development. Information and Software Technology, 54, 853-865.

MOE, N. B., DINGSØYR, T. & DYBÅ, T. 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. Information and Software Technology, 52, 480-491.

MUNN, P. & DREVER, E. 1990. Using Questionnaires in Small-Scale Research. A Teachers' Guide, ERIC.

NERUR, S. & BALIJEPALLY, V. 2007. Theoretical reflections on agile development methodologies. Communications of the ACM, 50, 79-83.

NERUR, S., MAHAPATRA, R. & MANGALARAJ, G. 2005. Challenges of migrating to agile methodologies. Communications of the ACM, 48, 72-78.

NEWSTED, P. R., HUFF, S. L. & MUNRO, M. C. 1998. Survey instruments in information systems. MIS quarterly, 22, 553-554.

OATES, B. J. 2005. Researching information systems and computing, Sage.

OPPENHEIM, A. N. 1992. Questionnaire design, interviewing and attitude measurement, Bloomsbury Publishing.

OVERHAGE, S. & SCHLAUDERER, S. Investigating the long-term acceptance of agile methodologies: An empirical study of developer perceptions in Scrum projects. System science (hicss), 2012 45th hawaii international conference on, 2012. IEEE, 5452-5461.

PALMER, S. R. & FELSING, M. 2001. A practical guide to feature-driven development, Pearson Education.

PAPATHEOCHAROUS, E., & ANDREOU, A. S. (2014). Empirical evidence and state of practice of software agile teams. Journal of Software: Evolution and Process, 26(9), 855-866.

PFEFFERMANN, D. A. R., C. 2011. Methods for design based sample Surveys, Burlington, MA., U.S.A., Elsevier.

PIKKARAINEN, M., SALO, O., KUUSELA, R. & ABRAHAMSSON, P. 2012. Strengths and barriers behind the successful agile deployment—insights from the three software intensive companies in Finland. Empirical software engineering, 17, 675-702.

PINSONNEAULT, A. & KRAEMER, K. L. 1993. Survey research methodology in management information systems: an assessment. Journal of management information systems, 75-105.

POPPENDIECK, M. & POPPENDIECK, T. 2003. Lean software development: an agile toolkit, Addison-Wesley Professional.

QUALTRICS 2013 Who uses Qualtrics
[Online] Available from:
http://www.qualtrics.com/about [Accessed: 6 June 2013]

REICH, M.2012 Agile vs. Waterfall: How to Approach your Web Development Project [Online] Available from:
http:/www.commonplaces.com/blog/agile-vs-waterfall-how-approach-your-web-development-project [Accessed: 3rd December 2012]

RETTIG, M. & SIMONS, G. 1993. A project planning and development process for small teams. Communications of the ACM, 36, 45-55.

ROBIN, D. P., NASH, H. W. & JONES, S. R. 1973. An Analysis of Monetary Incentives in Mail Questionnaire Studies. Journal of Business Communication, 11, 38-42.

ROYCE, W. 1970. Managing the Development of Large Software Systems: Concepts and Techniques, in 'Technical Papers of Western Electronic Show and Convention (WesCon)' .

SALO, O. & ABRAHAMSSON, P. 2008. Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. Software, IET, 2, 58-64.

SAUNDERS, M. N., SAUNDERS, M., LEWIS, P. & THORNHILL, A. 2011. Research methods for business students, 5/e, Pearson Education India.

SCHATZ, B. & ABDELSHAFI, I. 2005. Primavera Gets Agile. IEEE software, 3, 7.

SCHVEY, N., PUHL, R., LEVANDOSKI, K. & BROWNELL, K. 2013. The influence of a defendant's body weight on perceptions of guilt. International Journal of Obesity, 37, 1275-1281.

SCHWABER, K. & BEEDLE, M. 2002. gilè Software Development with Scrum.

SHAHIR, H. Y., DANESHPAJOUH, S. & RAMSIN, R. Improvement strategies for agile processes: a SWOT analysis approach. Software Engineering Research, Management and Applications, 2008. SERA'08. Sixth International Conference on, 2008. IEEE, 221-228.

SHARP, H. & ROBINSON, H. 2004. An ethnographic study of XP practice. Empirical Software Engineering, 9, 353-375.

SIDKY, A., ARTHUR, J. & BOHNER, S. 2007. A disciplined approach to adopting agile practices: the agile adoption framework. Innovations in systems and software engineering, 3, 203-216.

SIVO, S. A., SAUNDERS, C., CHANG, Q. & JIANG, J. J. 2006. How low should you go? Low response rates and the validity of inference in IS questionnaire research. Journal of the Association for Information Systems, 7, 17.

SLIGER, M. & BRODERICK, S. 2008. The software project manager's bridge to agility, Boston, MA. U.S.A, Addison-Wesley Professional.

SMITH, C. & KING, P. Agile Project Experiences The Story of Three Little Pigs. Agile, 2008. AGILE'08. Conference, 2008. IEEE, 378-383.

STAPLETON, J. 2003. DSDM: Business focused development, Harlow, Essex, Pearson Education.

STELLMAN, A. and GREENE, J. 2014 Learning Agile: Understanding Scrum, XP, Lean and Kanban, O'Reilly Media.

TINSLEY, H. E. & BROWN, S. D. 2000. Handbook of applied multivariate statistics and mathematical modeling, Academic Press.

VERSIONONE 2013 Agile Software Development Benefits
[Online] Available from:
http://www.versionone.com/agile-101/agile-software-development-benefits [Accessed: 8 March 2013]

VERSIONONE 2014 VersionOne Releases 9th Annual State of Agile Survey Results
[Online] Available from:
http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf [Accessed: 15 June 2014]

VIDGEN, R. & WANG, X. 2009. Coevolving systems and the organization of agile software development. Information Systems Research, 20, 355-376.

VIJAYASARATHY, L. & TURK, D. 2008. Agile Software Development: A survey of early adopters. Journal of Information Technology Management, 19, 1-8.

VLAANDEREN, K., JANSEN, S., BRINKKEMPER, S. & JASPERS, E. 2011. The agile requirements refinery: Applying Scrum principles to software product management. Information and Software Technology, 53, 58-70.

WALSHAM, G. 1995. The emergence of interpretivism in IS research. Information systems research, 6, 376-394.

WANG, X., CONBOY, K. & PIKKARAINEN, M. 2012. Assimilation of agile practices in use. Information Systems Journal, 22, 435-455.

WATERS, K. 2013. 10 Good Reasons To Do Agile Development
[Online] Available from:
http://www.allaboutagile.com/10-good-reasons-to-do-agile-development [Accessed: 8 March 2013]

WILLIAMS, L. 2012. What agile teams think of agile principles. Communications of the ACM, 55, 71-76.

ZOLYAK, A. 2013. 8 Benefits of Agile Software Development. [Online] Available from: http://www.seguetech.com/blog/2013/04/12/8-benefits-of-agile-software-development [Accessed: 5 May 2013]

Appendix A - Questionnaire Form

The following was sent out as a hardcopy to Agile practitioners, a web survey

https://bedsbusiness.eu.qualtrics.com/SE/?SID=SV_cLQYRO11uI9jgup

contained the same questions.

Agile Questionnaire

I am a research student at the University of Bedfordshire, and am investigating how the Agile software methodology is used in software development teams. As part of this research I would like you to assist me with thus study by completing the following questionnaire

The first section of the questions will establish your role, and how much of Agile you use (the level of adoption of different bits of Agile varies from organisation to organisation).

The second section asks what you like and dislike about Agile.

The aim of this questionnaire is to identify areas of Agile which are viewed as benefits and challenges by its users. Further research will then identify how the problem areas can be improved. The results will then be shared with everyone who completes the survey and leaves contact details, and with those organisations which have assisted with the study.

All of the information used in this survey will be collected anonymously unless you wish to leave contact details, and the questionnaires will be destroyed after the research project has been completed.

Alistair Streek

Alistair.Streek@beds.ac.uk

HOW YOU USE AGILE

1. What is your role in the team?

Developer	Tester	Analyst	Project Manager	Consultant	Visionary	Customer	Other
☐	☐	☐	☐	☐	☐	☐	☐

If Other please specify _____

2. How long have you been using Agile?

_____ years _____ months

3. Which Agile method do you use?

Not sure	Scrum	Kanban	DSDM	XP	Crystal	AUP	FDD	Lean	Hybrid	Other
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

If Other please specify _____

4. Does your team use the following? Please tick whichever applies.

	Always	Usually	Sometimes	Rarely	Never	Not Sure
Daily Scrum Meeting	☐	☐	☐	☐	☐	☐
Task Radiator	☐	☐	☐	☐	☐	☐
Backlog Queue	☐	☐	☐	☐	☐	☐
Stories	☐	☐	☐	☐	☐	☐
Sprint Planning	☐	☐	☐	☐	☐	☐
Sprints (short development cycles)	☐	☐	☐	☐	☐	☐
Increments (groups of Sprints)	☐	☐	☐	☐	☐	☐
Retrospectives (post-sprint sessions)	☐	☐	☐	☐	☐	☐
Pair Programming	☐	☐	☐	☐	☐	☐
Story Pointing	☐	☐	☐	☐	☐	☐
Must/Should/Could/Won't for stories	☐	☐	☐	☐	☐	☐

5. Does your team use an automated test tool?

Yes	No	Only on some platforms	Not sure
☐	☐	☐	☐

If yes, or on some platforms, what tools do you use? _____

6. Do you perform continuous integration by updating your build several times a day, as components are test complete?

Yes	No	Not sure
☐	☐	☐

WHAT DO YOU LIKE AND DISLIKE ABOUT AGILE?

7. What works well for you with Agile?

8. Which of the following are benefits of using Agile?

	Not a benefit	Not Significant 1	2	3	4	Very Significant 5
Less Documentation	☐	☐	☐	☐	☐	☐
Closer collaboration with team	☐	☐	☐	☐	☐	☐
Being able to see working software quickly	☐	☐	☐	☐	☐	☐
Can change the deliverable quickly	☐	☐	☐	☐	☐	☐
Working directly with end users	☐	☐	☐	☐	☐	☐

Know exactly what everyone is doing	☐	☐	☐	☐	☐	☐
Everyone on the team has an input to the design	☐	☐	☐	☐	☐	☐
Customer involvement	☐	☐	☐	☐	☐	☐
Bugs are found earlier	☐	☐	☐	☐	☐	☐
Reduced Risk	☐	☐	☐	☐	☐	☐
I like the fixed cost and variable features model	☐	☐	☐	☐	☐	☐
Developers empowered to manage their own development	☐	☐	☐	☐	☐	☐
Shortened Feedback Loop	☐	☐	☐	☐	☐	☐
Code gets to test earlier due to small deliverable size	☐	☐	☐	☐	☐	☐
Predictable communication between team members	☐	☐	☐	☐	☐	☐
Easy to integrate new staff with Agile experience	☐	☐	☐	☐	☐	☐
Focus on Must have items through using MSCW	☐	☐	☐	☐	☐	☐

Any other benefits? (Please specify)

	☐	☐	☐	☐	☐	☐
	☐	☐	☐	☐	☐	☐
	☐	☐	☐	☐	☐	☐

9. What do you find difficult about using Agile?

10. What disadvantages do you find with Agile?

	Not a problem	Not Significant 1	2	3	4	Very Significant 5
No history of what has been done	☐	☐	☐	☐	☐	☐
Working from very brief requirements	☐	☐	☐	☐	☐	☐
Less personal accountability as nothing is signed off	☐	☐	☐	☐	☐	☐
Use of Stories breaks continuity	☐	☐	☐	☐	☐	☐
Work is too time driven	☐	☐	☐	☐	☐	☐
No catering for story dependencies	☐	☐	☐	☐	☐	☐
Lack of an overview of what is to be done	☐	☐	☐	☐	☐	☐
Difficulties of using staff inexperienced in Agile methods	☐	☐	☐	☐	☐	☐
No story equals no coding mentality delays development	☐	☐	☐	☐	☐	☐
Hard to estimate work required for a large Agile project	☐	☐	☐	☐	☐	☐
Stories and Acceptance Criteria are not detailed enough	☐	☐	☐	☐	☐	☐
Always deliver the minimum set of features	☐	☐	☐	☐	☐	☐
Difficult to support after delivery for lack of documentation	☐	☐	☐	☐	☐	☐
Agile doesn't work without competent motivated people	☐	☐	☐	☐	☐	☐

Sprints are too short to deliver meaningful features	☐	☐	☐	☐	☐	☐

Any other disadvantages? (Please specify)

_____	☐	☐	☐	☐	☐	☐
_____	☐	☐	☐	☐	☐	☐
_____	☐	☐	☐	☐	☐	☐

11. Do you like using Agile?

Always	Usually	Occasionally	Rarely	Never
☐	☐	☐	☐	☐

12. How many employees does your organisation employ?

1 - 50	51 - 100	101 - 250	>250	Not sure
☐	☐	☐	☐	☐

Thank you very much for taking the time to answer this questionnaire.

Your response will be confidential, and all survey forms will be destroyed after the survey results have been compiled.

I would be interested in discussing the results of the survey with you, would you be willing to have a short discussion via Skype/Telephone/email to help me understand your use of Agile further?

No ☐ Yes ☐

My email is _____@_____

OR my skype ID is _____

OR my telephone number is _____

Appendix B - Structured enrichment questions

1. Many practitioners say that reduced documentation is a significant benefit of Agile. However this has appeared on the survey as the lowest rated benefit. Do you find that you produce less documentation with Agile that with Waterfall?
Or do you thing that having less documentation is a benefit but not a tangible one?
How do you support your system after release?

2. Use of Agile features – You have indicated that you have a daily Scrum, but you do not use a task radiator. How do you keep track of your progress? If you don't use a task radiator, would you consider using one in the future?

3. Do you use MSCW prioritisation?
If you do, what proportion of your delivered stories are usually Must haves? If you generally only deliver Must haves then does the code lack the validation and processes which make the software intuitive and easy to use? If not, how do you prioritise your stories?

4. The perception of users is that it is impossible to make Agile work without competent and motivated people. How easy did you find it to move people in your teams onto Agile? Were they resistant to the change, or enthusiastic from the beginning? How did you deal with those who became blockers to the new way of working?

5. That it is hard to estimate Agile projects is the biggest challenge the respondents to the survey have highlighted. Do you find that this is an issue for Agile projects? How do you estimate projects and how accurate have you found these estimations?

6. Once software has been released, how do you support it. What documentation is produced and how is this communicated? The use of Scrum meetings and retrospectives appears to reduce problems with supporting released software, has this been your experience?

Questions to targeted groups

To Company Two; The Task Radiator or Card Wall has been shown by the study to be a significant cause of people liking Agile. You do not use one - would you consider doing so in the future? How does your team publish progress?

To both Companies One and Two only: Stories not being detailed enough’ scores highly at TP while 'working from too brief requirements scores highly at Capita. Why would one be scored more highly that the other?

To Company Two: Do you use pair programming? Pair programming appears to be very popular amongst those using it. Do you find that using pair programming in a SCURM/Kanban environment works well? What challenges did you find when you originally started to use it?
If not, why has your organisation decided not to use it?

To Company One: As an organisation you attach a lower level of importance to Customer involvement in the development process than other organisations. Can you explain why this might be the case.

To Company One: You show an unusually low level of problems with “Difficulties using staff inexperienced in Agile”, why do you think this is?

Appendix C- Features by Challenges and Benefits Groups using Pearson Correlation

	BENEFITS					CHALLENGES		
	People_Team_Collaboration	People_Cust_Collaboration	Methodology_Speed_of_Release	Methodology_Process_Management	Easy to integrate new staff with Agile experience	People_Related	Methodology_Process_Mgmt	Methodology_Feature_Delivery
Increments (groups of Sprints)	-.007	.031	.092	-.085	-.013	.109	.041	.069
Retrospectives (post Sprint review sessions)	-.135	-.055	-.052	-.014	.106	.112	.090	.101
Pair Programming	-.189*	-.374**	-.044	-.231*	-.087	-.013	.052	.109
Sprints (short development cycles)	.046	.055	.096	.079	.114	.020	.029	-.038
Story Pointing	-.028	.035	-.041	-.039	-.020	.164	.070	.152
Must/Should/Could/Won't for stories	-.067	.322**	-.034	-.071	-.038	-.024	-.166	-.219*
Daily Scrum Meeting	-.048	.045	.073	.075	.056	.023	.076	.038
Task Radiator	-.104	-.020	-.129	-.180	.003	.260**	.198*	.141
Backlog Queue	-.023	.117	.070	.021	.153	.137	.122	.044
Stories	-.030	.210*	.049	.052	-.088	-.038	-.076	-.091
Sprint Planning	.087	.003	.142	.014	.142	.064	.071	-.024

**. Correlation is significant at the 0.01 level (2-tailed).*. Correlation is significant at the 0.05 level (2-tailed).

Appendix D – Correlation Tables

Features vs Benefits Pearson Correlations – part 1

	Less Documentation	Closer collaboration with team	Able to see working software quickly	Can change the deliverable quickly	Working directly with end users	Know exactly what everyone else is doing	Everyone on the team has an input to the design	Customer involvement	Bugs are found earlier
Daily Scrum Meeting	.082	.139	.101	-.010	.026	-.152	.015	.055	.141
Task Radiator	-.038	-.096	-.123	-.276**	-.009	-.118	-.072	-.026	-.065
Backlog Queue	.055	.103	.075	.049	.097	-.107	-.027	.116	.117
Stories	-.013	.087	.069	.006	.213*	-.074	-.008	.171	.107
Sprint Planning	-.020	.056	.142	.119	.077	-.011	-.004	-.070	.119
Sprints	.019	.004	.077	.030	.084	.019	-.014	.017	.100
Increments	-.026	.151	.096	-.019	.025	-.045	-.002	.032	.077
Retrospectives	.029	-.059	.083	.013	-.063	-.132	-.101	-.038	-.014
Pair Programming	-.199*	-.106	-.009	-.089	-.387**	-.262**	-.143	-.297**	-.043
Story Pointing	-.009	.083	.151	-.002	-.023	-.049	.049	.086	-.038
M/S/C/W	.143	.063	-.176	.056	.284**	-.092	-.062	.305**	-.102

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Features vs Benefits Pearson Correlations – part 2

	Reduced Risk	I like the fixed cost and variable features model	Developers empowered	Shortened feedback loop	Code gets to test earlier due to small deliverable size	Predictable communication between team members	Easy to integrate new staff with Agile experience	Focus on Must have items through MSCW
Daily Scrum Meeting	-.097	.029	.157	-.024	-.009	-.082	.056	.080
Task Radiator	-.194*	-.077	.026	-.147	-.067	-.029	.003	-.136
Backlog Queue	-.013	-.111	.117	-.019	.026	.002	.153	-.001
Stories	-.010	.107	.168	.055	-.057	-.052	-.088	-.075
Sprint Planning	-.056	-.025	.065	.021	.159	.193*	.142	-.009
Sprints	-.013	.047	.134	.022	.093	.107	.114	.059
Increments	-.051	-.061	.067	.045	.050	-.065	-.013	-.197*
Retrospectives	-.106	-.077	.073	-.139	-.097	-.090	.106	.015
Pair Programming	-.267**	-.141	-.283**	-.079	-.017	-.039	-.087	.117
Story Pointing	-.203*	.031	.031	-.065	-.149	-.119	-.020	-.013
M/S/C/W	.106	.007	.086	.183	-.039	-.063	-.038	-.578**

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Features vs Challenges Pearson Correlations – part 1

	No history of what has been done	Working from very brief requirements	Less personal accountability as nothing is signed off	Use of stories breaks continuity	Work is too time driven	No catering for story dependencies	Lack of an overview of what is to be done	Difficulties of using staff inexperienced in Agile methods
Daily Scrum Meeting	-.045	.069	.018	-.088	.053	-.008	-.040	-.085
Task Radiator	.076	.241*	.140	.015	.176	.145	.108	.147
Backlog Queue	.006	.091	.077	-.020	.030	-.012	.070	.021
Stories	-.153	-.079	.079	.019	-.031	-.068	-.273**	-.042
Sprint Planning	.011	.093	.124	.026	.048	-.081	-.124	-.029
Sprints	-.074	.078	.113	-.022	.035	-.108	-.121	-.074
Increments	-.095	-.083	.066	.026	.074	.075	.072	.087
Retrospectives	.018	.010	.108	.041	.030	-.011	-.016	-.005
Pair Programming	.049	.115	.088	.005	.083	.078	-.090	-.202*
Story Pointing	.091	.019	.190*	.068	-.018	.022	-.066	.069
Must/Should/Could/Won't	-.108	-.227*	.012	-.081	-.117	-.093	-.049	.135

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Features vs Challenges Pearson Correlations – part 2

	'No story equals no coding' mentality delays development	Hard to estimate work required for a large Agile project	Stories and acceptance criteria are not detailed enough	Always deliver the minimum set of features	Difficult to support after delivery due to lack of documentation	Agile doesn't work without competent motivated people	Sprints are too short to deliver meaningful features
Daily Scrum Meeting	.035	.192*	.055	.072	.271**	.048	-.117
Task Radiator	.131	.108	.159	.187	.207*	.247*	.040
Backlog Queue	.149	.244*	.104	.100	.237*	.102	-.147
Stories	-.174	.084	-.006	-.155	.028	.009	.089
Sprint Planning	.018	.198*	.068	.017	.193*	.047	-.117
Sprints	-.023	.175	.072	.056	.138	.094	-.111
Increments	.057	.196*	-.137	.033	.133	.046	.036
Retrospectives	.112	.165	.037	.141	.278**	-.028	-.138
Pair Programming	.017	.116	-.112	.164	.092	.025	-.056
Story Pointing	.005	.132	-.034	.221*	.215*	.089	-.042
Must/Should/Could/Won't	-.086	-.047	-.141	-.283**	-.172	-.078	-.172

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

Appendix E - Features, Benefits and Challenges by company

Feature Correlation by Company

Independent Samples Test

Comparison of Company One and Company Two Equal variances assumed	Levene's Test for Equality of Variances		t-test for Equality of Means							
								95% Confidence Interval of the Difference		
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper	Eta Squared
Daily Scrum Meeting	8.400	.006	1.317	42	.195	.083	.063	-.044	.211	
Task Radiator	14.369	.000	-1.200	42	.237	-.725	.604	-1.944	.494	
Backlog Queue	11.708	.001	-1.603	42	.117	-.525	.328	-1.186	.136	
Sprint Planning	8.240	.006	1.309	42	.198	.375	.287	-.203	.953	
Sprints	3.629	.064	.911	42	.367	.208	.229	-.253	.670	
Increments	31.493	.000	-2.276	42	.028	-.750	.330	-1.415	-.085	
Retrospectives	37.013	.000	2.241	42	.030	.208	.093	.021	.396	0.10684
Pair Programming	10.081	.003	5.491	42	.000	2.358	.429	1.492	3.225	0.41791
Story Pointing	.457	.503	-.095	42	.925	-.033	.352	-.744	.677	
MSCW	41.622	.000	-10.193	42	.000	-3.008	.295	-3.604	-2.413	0.71214

Challenges Correlation by Company

Equal variances assumed	Levene's Test for Equality of Variances		t-test for Equality of Means							
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference		Eta Squared
								Lower	Upper	
No history of what has been done	.005	.946	-.742	42	.462	-.43333	.58422	-1.61234	.74567	
Working from very brief requirements	1.952	.170	-.093	42	.926	-.05000	.53613	-1.13196	1.03196	
Less personal accountability as nothing is signed off	.612	.439	-1.039	42	.305	-.53333	.51343	-1.56948	.50282	
Use of stories breaks continuity	.410	.526	.031	42	.976	.01667	.54006	-1.07321	1.10654	
Work is too time driven	1.380	.247	.916	42	.365	.48333	.52779	-.58179	1.54846	
No catering for story dependencies	1.589	.214	-.385	42	.702	-.20000	.51907	-1.24753	.84753	
Lack of an overview of what is to be done	.801	.376	-2.284	42	.027	-1.13333	.49614	-2.13458	-.13208	0.11051
Difficulties of using staff inexperienced in Agile methods	.094	.760	-2.752	42	.009	-1.29167	.46941	-2.23897	-.34436	0.152743
'No story equals no coding' mentality delays development	.321	.574	-.581	42	.564	-.31667	.54458	-1.41568	.78235	

Hard to estimate work required for a large Agile project	.150	.701	-.742	42	.462	-.40000	.53888	-1.48749	.68749	
Stories and acceptance criteria are not detailed enough	5.921	.019	-2.482	42	.017	-1.32500	.53384	-2.40233	-.24767	0.127914
Always deliver the minimum set of features	.443	.509	.740	42	.463	.41667	.56272	-.71894	1.55228	
Difficult to support after delivery due to lack of documentation	3.221	.080	-.151	42	.881	-.08333	.55096	-1.19521	1.02855	
Agile doesn't work without competent motivated people	.077	.783	-.878	35	.386	-.45294	.51615	-1.50077	.59489	
Sprints are too short to deliver meaningful features	.613	.441	.278	26	.783	.13333	.48008	-.85349	1.12016	

Benefits Correlation by Company

Equal variances assumed	Levene's Test for Equality of Variances		t-test for Equality of Means							
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference		Eta Squared
								Lower	Upper	
Less Documentation	.037	.847	-1.984	42	.054	-.89167	.44934	-1.79846	.01513	
Closer collaboration with team	.492	.487	-.343	42	.733	-.08333	.24313	-.57398	.40732	
Able to see working software quickly	1.078	.305	1.248	42	.219	.55000	.44061	-.33919	1.43919	
Can change the deliverable quickly	.528	.472	1.222	42	.229	.40833	.33415	-.26601	1.08267	
Working directly with end users	2.812	.101	-4.636	42	.000	-1.75833	.37927	-2.52374	-.99293	0.33851
Know exactly what everyone else is doing	.725	.399	-.331	42	.742	-.12500	.37773	-.88730	.63730	
Everyone on the team has an input to the design	.118	.733	.168	42	.868	.06667	.39747	-.73545	.86879	
Customer involvement	20.009	.000	-4.013	42	.000	-1.57500	.39247	-2.36704	-.78296	
Bugs are found earlier	.029	.867	1.511	42	.138	.61667	.40803	-.20678	1.44011	
Reduced Risk	1.012	.320	-.468	42	.642	-.20833	.44494	-1.10627	.68960	
I like the fixed cost and variable features model	.444	.509	.949	42	.348	.46667	.49172	-.52567	1.45900	
Developers empowered	2.061	.159	-.700	42	.488	-.31667	.45266	-1.23018	.59684	
Shortened feedback loop	1.487	.229	-.359	42	.722	-.10833	.30189	-.71758	.50091	

Code gets to test earlier due to small deliverable size	.530	.471	.872	42	.388	.30000	.34394	-.39410	.99410	
Predictable communication between team members	1.651	.206	.894	42	.377	.38333	.42890	-.48221	1.24888	
Easy to integrate new staff with Agile experience	.436	.513	1.620	42	.113	.69167	.42692	-.16989	1.55323	
Focus on Must have items through MSCW	14.960	.000	3.157	42	.003	1.29167	.40917	.46593	2.11740	